Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа по

Основам программной инженерии №1

Вариант 3094

Работу выполнили:

Гаврилин О.С.

Зорин Г.Ю.

Группа:

P3230

Санкт-Петербург,

2025

Оглавление

[Task 3](#_Toc192676740)

[1. Introduction 4](#_Toc192676741)

[2. General description 5](#_Toc192676742)

[3. **Functional requirements** 6](#_Toc192676743)

[**4. Nonfunctional requirements** 8](#_Toc192676744)

[**5. Requirements attributes** 9](#_Toc192676745)

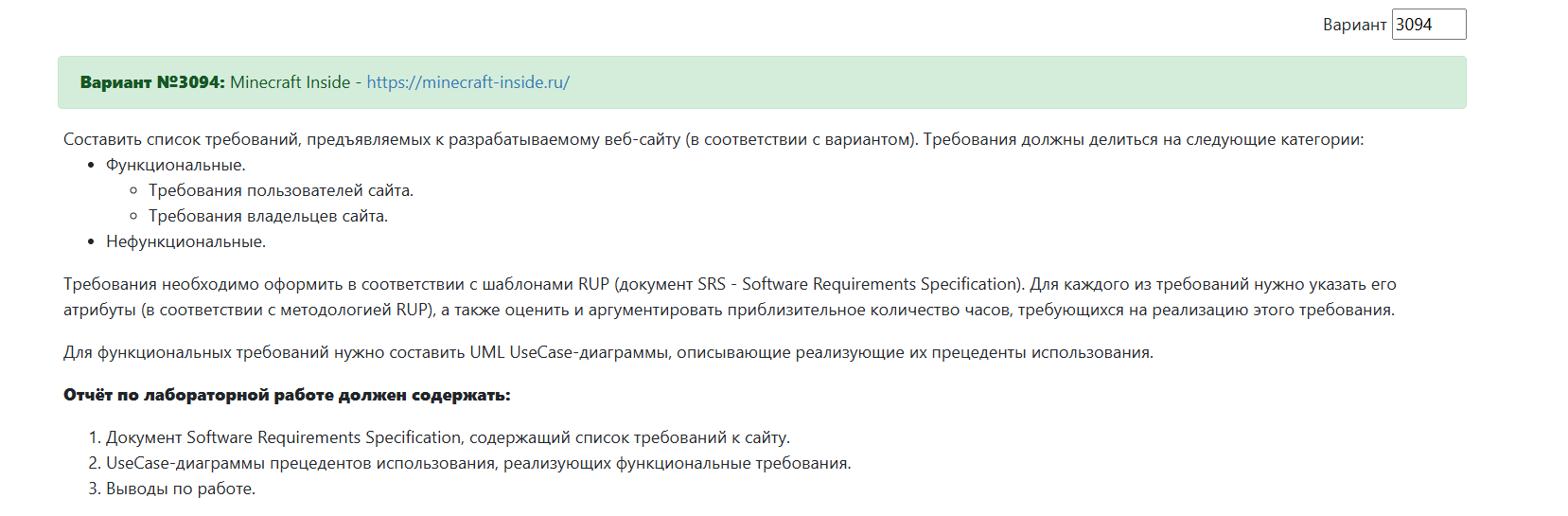
[**6. Precedents** 11](#_Toc192676746)

[**7. Mockups** 13](#_Toc192676747)

[**8. Use-case diagrams** 16](#_Toc192676748)

[**Вывод** 17](#_Toc192676749)

## Task



Software Requirements Specification

### 1. Introduction

#### 1.1 Purpose of the document

Документ описывает и формулирует функциональные и нефункциональные требования для разработки системы. Документ можно рассматривать как соглашение между заказчиком и исполнителем о продукте, который должен быть разработан.

#### 1.2 Field of application

Документ относится к веб-сайту <https://minecraft-inside.ru/> - платформе, обеспечивающей возможность пользователям публиковать, просматривать оценивать и обсуждать различные материалы по игре minecraft.

#### 1.3 Definitions, abbreviations and abbreviations

* SRS (Software Requirements Specification) – Спецификация требований к программному обеспечению.
* UML (Unified Modeling Language) – Унифицированный язык моделирования, используемый для разработки диаграмм, таких как Use-Case.
* Use-Case Diagram – Диаграмма прецедентов, отображающая взаимодействие пользователей с системой.
* CAPTCHA – Технология для проверки, что пользователь является человеком, а не автоматической программой.
* PHP – Язык программирования, используемый для разработки серверной части веб-приложений.
* MySQL – Реляционная система управления базами данных, используемая для хранения данных сайта.
* Redis/Memcached – Системы кэширования, используемые для улучшения производительности веб-приложений.
* HTTP/HTTPS – Протоколы передачи гипертекста; HTTPS обеспечивает безопасную передачу данных.

#### 1.4 Document overview

Этот документ представляет собой спецификацию требований к веб-сайту Minecraft Inside, который служит платформой для публикации, скачивания, оценки и обсуждения материалов, связанных с игрой Minecraft. Документ включает в себя как функциональные, так и нефункциональные требования, которые будут использоваться для разработки и поддержания системы. Также в документе описаны роли пользователей, их права и обязанности, а также процессы, которые должны быть реализованы для обеспечения нормальной работы сайта. Документ ориентирован на всех участников проекта, включая заказчиков, разработчиков и тестировщиков, и служит основой для реализации и тестирования системы.

### 2. General description

#### 2.1 User roles

* **Неавторизованный пользователь –** имеет доступ к просмотру всех материалов, опубликованных другими пользователями, имеет возможность скачивать опубликованные другими пользователями материалы, имеет доступ к перемещению по всему веб сайту.
* **Авторизованный пользователь –** обладает возможностями неавторизованного пользователя (см. предыдущий пункт), а также: публиковать материалы на веб-сайте, оценивать публикации других пользователей, комментировать публикации других пользователей, добавлять материалы в закладки, настраивать уведомления, редактировать данные своего профиля.
* **Модератор –** обладает правами на удаление комментариев, учетных записей, пользовательских публикаций. Обладает возможностью: публиковать материалы, получать доступ к метрикам веб-сайта. Обладает доступом к обработке пользовательских жалоб.
* **Администратор –** обладает набором прав модератора. Обладает полномочиями назначать модераторов системы. Обладает всеми метриками сайта, в праве распоряжаться ими.

#### Dependencies

1. Серверное окружение: Linux + Appache/Nginx, PHP 8+, MySQL 8+.
2. Поддержание механизма CAPTCHA

### 3. **Functional requirements**

#### **Unauthorized user**

* + 1. Система должна обеспечивать возможность скачивания материалов, опубликованных на веб-сайте.
    2. Система должна обеспечивать возможность регистрации учетной записи.
    3. Система должна предоставлять возможность аутентификации пользователя.
    4. Система должна предоставлять возможность авторизации пользователя.
    5. Система должна предоставлять возможность поиска требуемых материалов согласно доступным категориям.
    6. Система должна предоставлять возможность использование генератора для создания различных для создания необходимых ресурсов.
    7. Система должна предоставлять возможность просматривать форум (темы, сообщения, обсуждения)

#### **Authorized user**

Система должна предоставлять все функции, которыми обладает Unauthorized user (п. 3.3), а также:

* + 1. Система должна предоставлять пользователю возможность загружать материалы.
    2. Система должна обеспечивать хранение опубликованных пользователями материалов.
    3. Система должна предоставлять возможность оценки опубликованного другими пользователями контента.
    4. Система должна предоставлять возможность комментировать опубликованный на платформе материал.
    5. Система должна предоставлять возможность менять имя пользователя учетной записи.
    6. Система должна предоставлять возможность сбрасывать пароль учетной записи.
    7. Система должна предоставлять возможность устанавливать пароль учетной записи.
    8. Система должна обеспечивать хранение учетных данных пользователей.
    9. Система должна предоставлять возможность отправки жалобы на публикации других пользователей (например, на нарушение авторских прав).
    10. Система должна обеспечивать возможность добавлять материалы в закладки.

#### **Moderator**

* + 1. Система должна предоставлять возможность удалять комментарии пользователей.
    2. Система должна предоставлять возможность блокировать учетные записи пользователей.
    3. Система должна предоставлять возможность удалять материалы, опубликованные другими пользователями.
    4. Система должна предоставлять возможность отслеживать жалобы пользователей.
    5. Система должна предоставлять возможность модерации материалов (одобрение или отклонение публикации пользователей).

#### **Administrator**

Система должна обеспечивать все те же возможности, что и для роли Moderator (п. 3.3), а также:

* + 1. Система должна предоставлять возможность назначать пользователю роль модератора с обеспечением всех возможностей этой роли (п. 3.3)
    2. Система должна обеспечивать доступ к аналитике веб-сайта (статистика посещений, активность пользователей, загруженные материалы).

### **4. Nonfunctional requirements**

#### **Reliability**

* + 1. Система должна обеспечивать доступность не менее 99% времени в месяц.
    2. Система должна автоматически переключаться на резервный сервер в случае отказа основного.
    3. Время восстановления системы после сбоя не должно превышать 1 (одного) часа
    4. Система должна обеспечивать механизм уведомлений администратора о внутренних сбоях.
    5. Система должна использовать кэширование (Redis или Memcached).

#### **Performance**

* + 1. Система должна обеспечивать среднее время загрузки страницы, не более 3 секунд при средней скорости интернета 10 Мбит/с.
    2. Система должна обеспечивать бесперебойную работу при нагрузке до 5000 активных пользователей.
    3. Система должна обрабатывать до 600 запросов в секунду без значительного снижения производительности.
    4. Система должна производить резервное копирование всех данных раз в 24 часа.
    5. Максимальный размер скачиваемого из системы файло не должен превышать 100МБ.
    6. Максимальный размер загружаемого в систему файла не должен превышать 100МБ.

#### **Supportability**

* + 1. Пользовательский интерфейс должен корректно работать на устройствах с минимальным разрешением экрана 320×480 пикселей и выше
    2. Система должна корректно работать в браузерах следующих версий: Chrome 90+, Firefox 85+, Edge 90+, Safari 17+
    3. Все пользовательские данные система должна передавать посредством протокола HTTPS.
    4. Система должна поддерживать следующие форматы загружаемых и скачиваемых файлов: .zip, .rar, .mcworld, .mcpack, .jar.

### **5. Requirements attributes**

**Functional requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ требования** | **Приоритет** | **Трудоемкость человеко-час** | **Стабильность** |
| 3.1.1 | 9 | 100 | Высокая |
| 3.1.2 | 10 | 150 | Высокая |
| 3.1.3 | 10 | 50 | Высокая |
| 3.1.4 | 10 | 100 | Высокая |
| 3.1.5 | 8 | 5 | Средняя |
| 3.1.6 | 4 | 6 | Низкая |
| 3.1.7 | 9 | 75 | Высокая |
| 3.2.1 | 10 | 60 | Низкая |
| 3.2.2 | 10 | 45 | Высокая |
| 3.2.3 | 10 | 10 | Низкая |
| 3.2.4 | 10 | 60 | Средняя |
| 3.2.5 | 5 | 10 | Низкая |
| 3.2.6 | 8 | 5 | Высокая |
| 3.2.7 | 9 | 5 | Высокая |
| 3.2.8 | 10 | 80 | Высокая |
| 3.2.9 | 7 | 60 | Средняя |
| 3.2.10 | 8 | 30 | Низкая |
| 3.3.1 | 5 | 40 | Низкая |
| 3.3.2 | 7 | 20 | Средняя |
| 3.3.3 | 9 | 30 | Высокая |
| 3.3.4 | 10 | 50 | Высокая |
| 3.3.5 | 9 | 90 | Высокая |
| 3.4.1 | 6 | 5 | Низкая |
| 3.4.2 | 7 | 20 | Средняя |

**Nonfunctional requirements**

|  |  |  |
| --- | --- | --- |
| **№ требования** | **Приоритет** | **Стабильность** |
| 4.1.1 | 10 | Высокая |
| 4.1.2 | 8 | Средняя |
| 4.1.3 | 9 | Высокая |
| 4.1.4 | 6 | Низкая |
| 4.1.5 | 10 | Средняя |
| 4.2.1 | 5 | Средняя |
| 4.2.2 | 7 | Высокая |
| 4.2.3 | 9 | Средняя |
| 4.2.4 | 10 | Низкая |
| 4.2.5 | 7 | Средняя |
| 4.2.6 | 7 | Высокая |
| 4.3.1 | 7 | Низкая |
| 4.3.2 | 8 | Высокая |
| 4.3.3 | 6 | Высокая |

**Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Риск** | **Тип риска** | **Вероятность** | **Масштаб потерь** |
| 1 | DDoS-атака | Технический | Низкая | Средние |
| 2 | Перебой в работе хостинга | Технический | Низкая | Значительные |
| 3 | Нагрузка на сервер, превышающая ожидаемые показатели | Технический | Средняя | Средние |
| 4 | Задержка зарплат разработчикам | Ресурсный | Очень низкая | Средние |
| 5 | Проигрыш в конкуренции | Бизнес-риск | Средняя | Значительные |
| 6 | Недостаточная компетентность разработчиков | Технический | Средняя | Средняя |
| 7 | Нехватка людей для работы над проектом | Ресурсный | низкая | Средняя |

**Стратегии по предотвращению рисков**

1. DDoS – атака: Использование CDN для фильтрации подозрительных запросов. Внедрение WAF (Web Application Firewall) для блокировки вредоносных запросов. Ограничение частоты запросов от одного IP-адреса. Использование CAPTCHA при частых запросах (например, для незарегистрированных пользователей).

2. Перебой в работе хостинга: Использование резервного хостинга или облачных решений с автоматическим переключением (AWS, DigitalOcean). Настройка автоматического резервного копирования базы данных и файлов на сторонний сервер. Мониторинг доступности сервера с автоматическими уведомлениями в случае сбоя.

3. Предотвращение перегрузки сервера: Балансировка нагрузки между несколькими серверами. Хранение часто скачиваемых файлов через CDN. Оптимизация базы данных для уменьшения нагрузки.

4. Предотвращение проигрыша в конкуренции: Улучшение пользовательского интерфейса, добавление удобной системы поиска и рекомендаций. Развитие сообщества: привлечение моддеров, разработка эксклюзивного контента. Уникальное функции, а также получение обратной связи.

**Стратегии реагирования**

1. DDoS – атака: Переключение сайта на режим ограниченного доступа (только для зарегистрированных пользователей). Активация анти-DDoS-защиты у провайдера хостинга. Временное ограничение скачивания файлов до стабилизации ситуации. Анализ логов и блокировка IP-адресов злоумышленников.

2. Действия при сбое хостинга: Переключение на резервный сервер. Информирование пользователей через Telegram о причинах сбоя. Восстановление базы данных из резервной копии.

3. Действия при перезагрузки сервера: Временное отключение второстепенных функций (комментарии, рейтинг модов). Ограничение скорости скачивания файлов для незарегистрированных пользователей. Масштабирование инфраструктуры (временное использование облачных серверов).

4. Действия при проигрыше в конкуренции: Проведение опросов пользователей, чтобы понять, чего им не хватает. Улучшение системы рекомендаций (например, "Похожие моды", "Популярное за неделю"). Развитие дополнительных функций, которых нет у конкурентов.

### **6. Precedents**

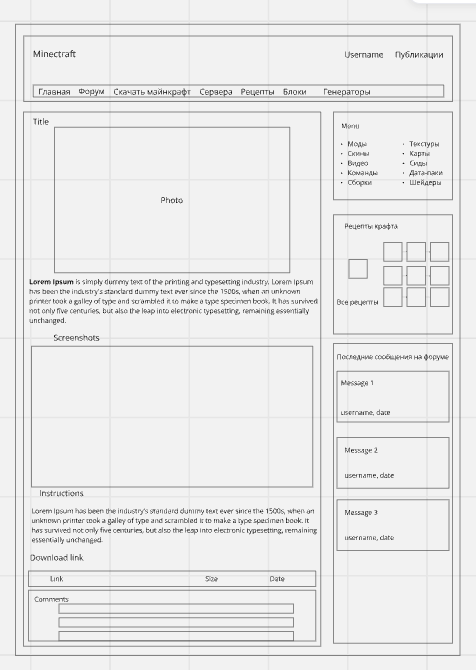
|  |
| --- |
| **Прецедент: Просмотр контента** |
| **Id**: 1 |
| **Краткое описание:** Пользователь просматривает контент на платформе |
| **Главный персонаж:** Авторизованный пользователь, Неавторизованный пользователь, Незарегистрированный пользователь |
| **Предусловия:** Пользователь находится на главной странице |
| **Основной поток:**   1. Пользователь выбирает необходимый контент, нажимая на его превью или заголовок 2. Система открывает выбранную страницу 3. На странице автоматически разворачивается описание опубликованных материалов 4. Пользователь может управлять показываемым контентом. |
| **Альтернативный поток:** Если контент недоступен (нет соединения, удалён модератором, доступ ограничен) система отображает соответствующее сообщение пользователю. |
| **Постусловия:** Пользователь успешно просмотрел необходимый контент. |

|  |
| --- |
| **Прецедент: Авторизация пользователя** |
| **Id:** 2 |
| **Краткое описание:** Пользователь проходит авторизацию в системе, получая доступ к дополнительному функционалу. |
| **Главный персонаж:** Неавторизованный пользователь |
| **Предусловия:** Пользователь находится на странице авторизации или на главной странице |
| **Основной поток:**   1. Пользователь нажимает кнопку входа в систему 2. Система предлагает ввести данные для входа 3. Пользователь вводит необходимые данные 4. Система авторизует пользователя и предоставляет ему доступ к дополнительному функционалу |
| **Альтернативный поток:** При отсутствии введенных данных пользователем в системе или некорректном вводе данных, система указывает на ошибку и предлагает ввести данные заново.  Если пользователь несколько раз ввёл неверный пароль, система может временно заблокировать попытки входа и потребовать качу. |
| **Постусловия:** После успешной авторизации пользователь получает доступ к дополнительному функционалу, система перенаправляет его на профиль или последнюю посещенную страницу. (если нажатие кнопки было не на главной странице) |

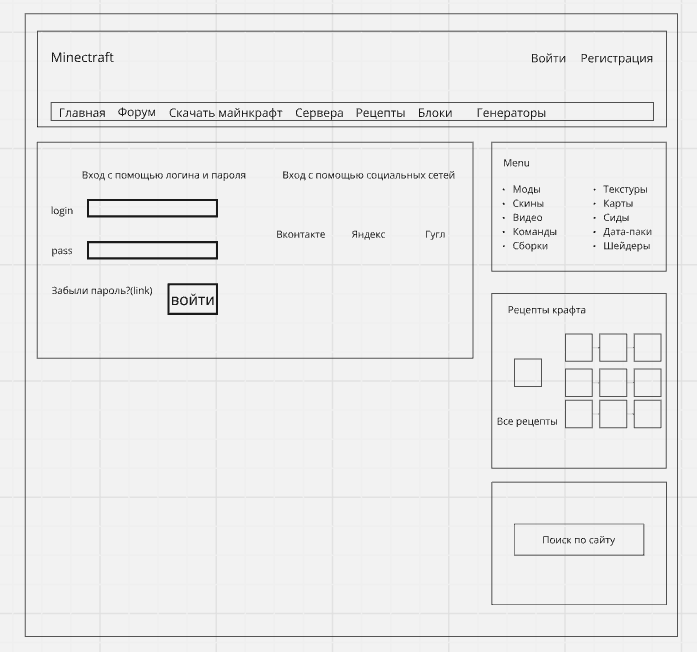
|  |
| --- |
| **Прецедент: Загрузка контента** |
| **Id:** 3 |
| **Краткое описание:** Авторизованный пользователь загружает материалы для их публикации на платформе |
| **Главный персонаж:** Авторизованный пользователь |
| **Предусловия:** Пользователь авторизован в системе |
| **Основной поток:**   1. Пользователь заходит в раздел “Мои публикации” 2. Пользователь нажимает кнопку “Добавить новость” 3. Заполняет все необходимые поля 4. Подтверждает, что ознакомлен с правилами публикации контента 5. Публикует материалы |
| **Альтернативный поток:** Если публикуемые материалы нарушают правила платформы, то публикация откланяется, и пользователь уведомляется об этом. Если загруженные материалы превышают максимально допустимый размер, то пользователю предлагается сократить их или сжать. |
| **Постусловия:** Контент опубликован на платформе, удален или не допущен к публикации с уведомлением автора и объяснением причин. |

|  |
| --- |
| **Прецедент: Изменение настроек учетной записи** |
| **Id:** 4 |
| **Краткое описание:** Пользователь изменяет настройки своей учетной записи на платформе |
| **Главный персонаж:** Авторизованный пользователь |
| **Предусловия:** Пользователь авторизован на платформе |
| **Основной поток:**   1. Пользователь заходит в раздел “Настройки” в своем профиле 2. Пользователь выбирает пункт, который хочет изменить в своем профиле 3. Пользовать вносит необходимые изменения в профиле 4. Пользователь сохраняет изменения |
| **Альтернативный поток:** в случае невозможности сохранения изменений по причине ошибки внутри системы пользователю предлагается повторить попытку позже. |
| **Постусловия:** Изменения внесены корректно или пользователь уведомлен о невозможности внесения/сохранения изменений |

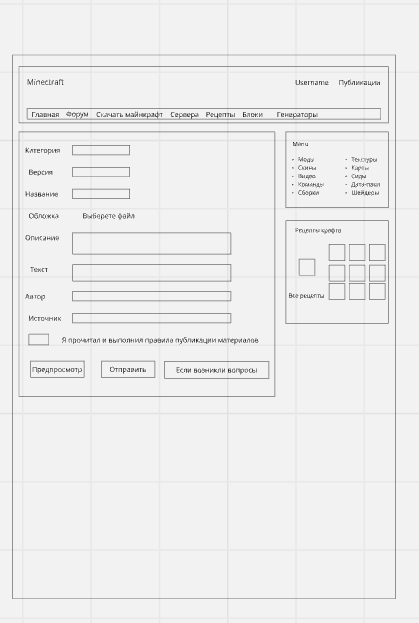
### **7. Mockups**



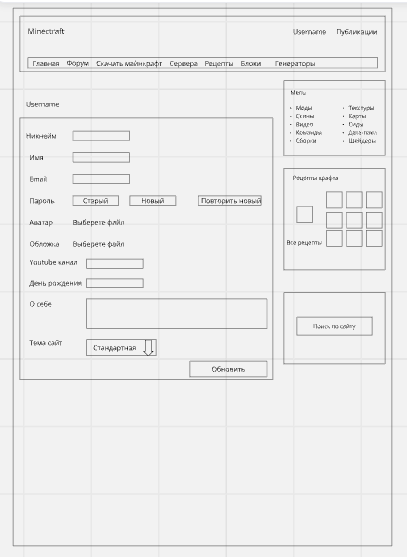
1. Просмотр контента на платформе



1. Authorization

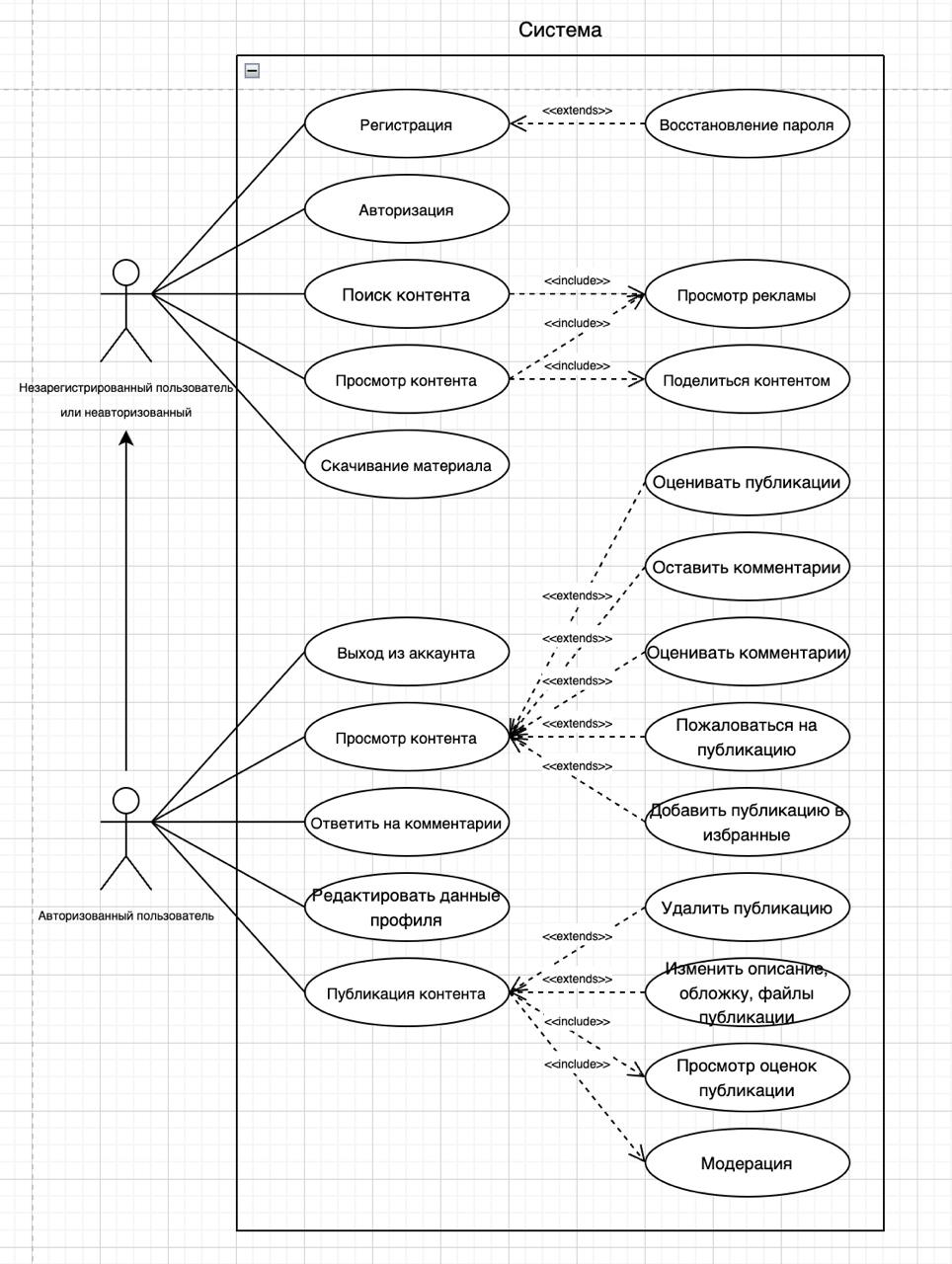


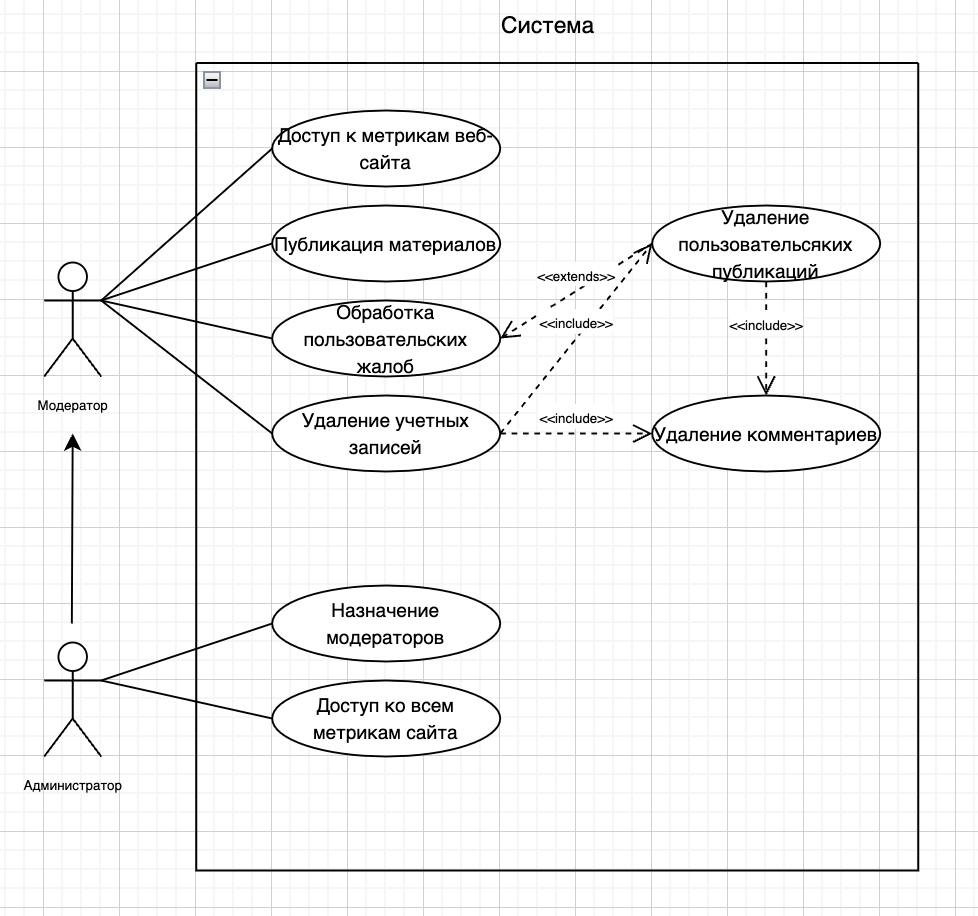
1. Content loading



1. Profile editing

### **8. Use-case diagrams**





### **Вывод**

В ходе выполнения лабораторной работы мы ознакомились со структурой SRS – документа и создали список требований, предъявляемых веб-сайту. Ознакомились с методологией RUP, проработали UML Use-Case – диаграммы на основе прецедентов их использования. Также успели попрактиковаться в расчете трудоемкости всех процессов и оценке рисков.