

Новосибирский Государственный Университет
Факультет Информационных Технологий

Техническое описание проекта по курсу ООАД

“NSUMedia бот”

**Студенты ФИТ НГУ
Петров Сергей
группа 21211**

**Студенты ФИТ НГУ
Авдеев Виталий
группа 21211**

Версия 1.0.0

Содержание

1. Введение	3
1.1 Цель	3
1.2 Область действия	3
1.3 Определения и сокращения	3
1.4 Ссылки	3
1.5 Краткое описание	3
2. Предметная область проекта	4
2.1 Существующие проблемы	4
2.2 Предлагаемое решение	4
3. Требования к программному решению	4
3.1 Роли	4
3.2 Функциональные требования для роли «Зритель материала»	5
3.2.1 Просмотр учебного материала	5
3.3 Функциональные требования для роли «Незарегистрированный пользователь»	8
3.3.1 Регистрация	8
3.4 Функциональные требования для роли «Зарегистрированный пользователь»	9
3.4.1 Активация аккаунта	10
3.5 Функциональные требования для роли «Зарегистрированный пользователь»	11
3.5.1 Предложение материала	11
3.6 Функциональные требования для роли «Администратор»	12
3.6.1 Принятие и отклонение материала, отправленного пользователем	13
3.6.2 Добавление материала	14
3.6.3 Удаления материала	14
3.7 Нефункциональные требования	15
4. Обзор архитектуры	15
4.1 Компонентная модель системы	16
4.1.1 Компонент 1	16
4.1.2 Компонент 2	16
4.2 Компоненты сторонних производителей	16
4.3 Схема развертывания приложения	16
5. Допущения и ограничения	16
6. Известные проблемы	17
6.1 Невысокая производительность приложения	17

Техническое описание проекта по курсу ООАД

1. Введение

1.1 Цель

Данный документ представляет собой техническое описание проекта "NSUMedia бот" и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

1.2 Область действия

Документ разработан в рамках проекта "NSUMedia бот" на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

1.3 Определения и сокращения

*[В этой таблице нужно перечислить все термины предметной области, используемые далее в документе. В тексте документа термины имеет смысл выделять курсивом. Текст, выделенный зеленым, является ПРИМЕРОМ, в вашем проекте он **может и должен** быть другим.]*

Таблица 1: Определения и сокращения

Термин	Описание
ATM	Automated Teller Machine - банкомат
VISA	Система пластиковых карт VISA

1.4 Ссылки

В тексте содержатся ссылки на следующие документы:

[1] diagrams.asta - UML диаграммы

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

1.5 Краткое описание

Содержание данного документа построено таким образом, чтобы дать ответ на следующие вопросы:

- Какие проблемы предметной области должен решать будущий программный продукт
- Посредством какой функциональности системы будут достигнуто решение проблем предметной области
- Какова архитектура программного решения

Описание предметной области и проблем, для решения которых предназначен будущий программный продукт, приведены в разделе 2.

Раздел 3 содержит описание требований к программному решению, раздел – описание архитектуры выбранного решения.

2. Предметная область проекта

Обучение и учебный процесс студентов требуют доступа к обширным объемам учебного материала, не всегда имеющего в библиотеках или общедоступных сайтах. Студенты всегда ищут удобные способы доступа к этой информации, чтобы успешно учиться и готовиться к экзаменам. Однако поиск, организация и сортировка учебных материалов может быть вызовом, а существующие платформы и ресурсы не всегда удовлетворяют потребности студентов.

2.1 Существующие проблемы

Студенты часто сталкиваются с неудобствами при поиске и доступе к необходимым учебным материалам. Ресурсы, предоставляемые учебными заведениями, могут быть неструктурированными, и важные материалы могут быть разбросаны по разным источникам. Это приводит к потере времени и снижению эффективности обучения. Кроме того, не всегда существует простой способ обновления или дополнения материалов, а также обмена ими между студентами.

2.2 Предлагаемое решение

Мы предлагаем создать Telegram-бота, который будет решать эти проблемы, предоставляя студентам удобный доступ к учебным материалам, организованным по курсам, семестрам и дисциплинам. Студенты смогут легко загружать, обновлять и обмениваться учебными материалами через этого бота. Наш бот станет надежным партнером студентов в учебном процессе, помогая им легко находить учебные ресурсы и управлять ими.

3. Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта "NSUMedia бот".

3.1 Роли

Основные роли и их иерархия описаны в *diagrams/1. Use-case модель/Основные UC^[1]* (см. Рис.1).

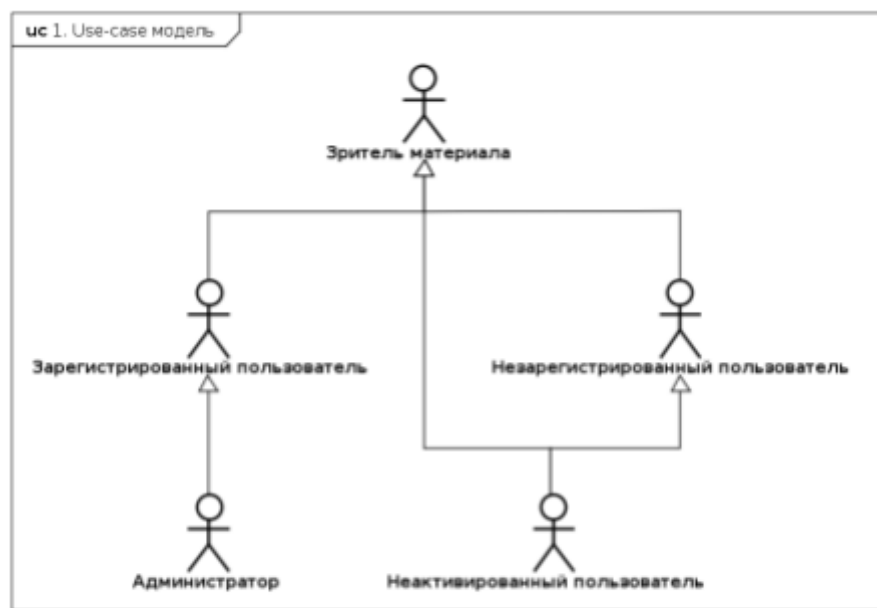


Рис.1. Иерархия ролей

1. **Зритель материала.** Зритель материала - фиктивный актёр, имеющий возможность просматривать учебный материал. От зрителя материала наследуются все остальные роли.
2. **Незарегистрированный пользователь.** Пользователь без регистрации может начать процесс регистрации и получить больше привилегий.

3. **Неактивированный пользователь.** Неактивированный пользователь - промежуточная роль между незарегистрированным и зарегистрированным пользователем. Возникает при начале регистрации и заканчивается после активации аккаунта. Имеет возможность активироваться. Также наследует возможности незарегистрированного пользователя, т.е. способен заново начать регистрацию.
4. **Зарегистрированный пользователь.** Зарегистрированный пользователь получает возможность предлагать новый материал, адресуя его администраторам.
5. **Администратор.** Администратор подписан на рассылку ботом нового материала для добавления, может принимать или отклонять его. Также имеет возможность добавлять или удалять материал напрямую.

3.2 Функциональные требования для роли «Зритель материала»

Use-case и их сценарии для роли «Зритель материала» описаны в *diagrams/1. Use-case модель/UC зрителя материала^[1]* (см. Рис. 2).

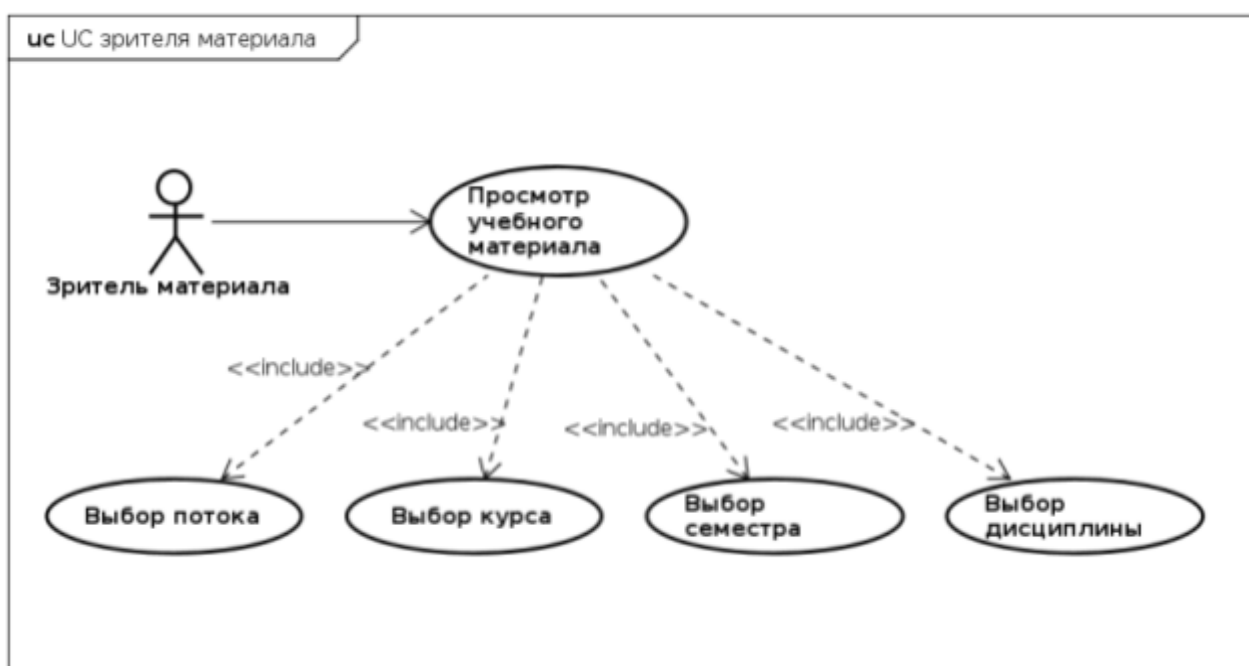


Рис. 2. Use cases Зрителя материала

3.2.1 Просмотр учебного материала

Описывает процесс просмотра учебного материала на платформе. Для удобства нахождения необходимого пособия Зритель материала последовательно выбирает ту или иную кнопку выбора потока/курса/семестра/дисциплины (см. Рис 3-7).

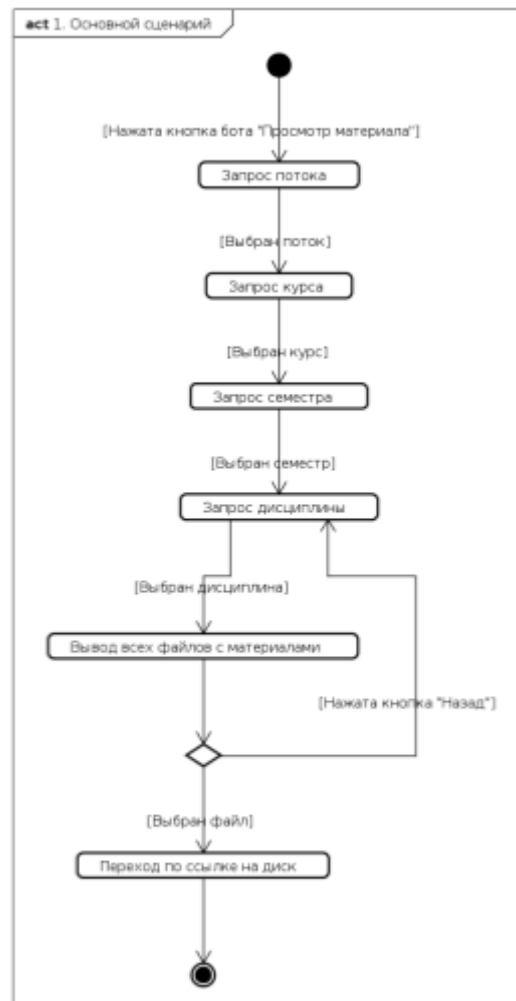


Рис. 3. Основной сценарий просмотра материала

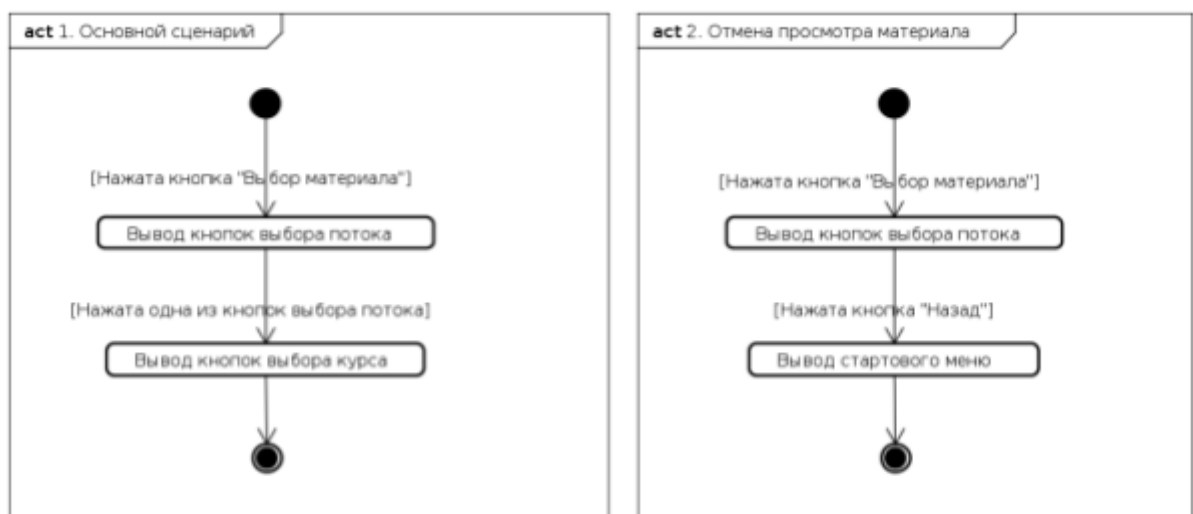


Рис. 4. Основной сценарий при выборе потока и сценарий отмены

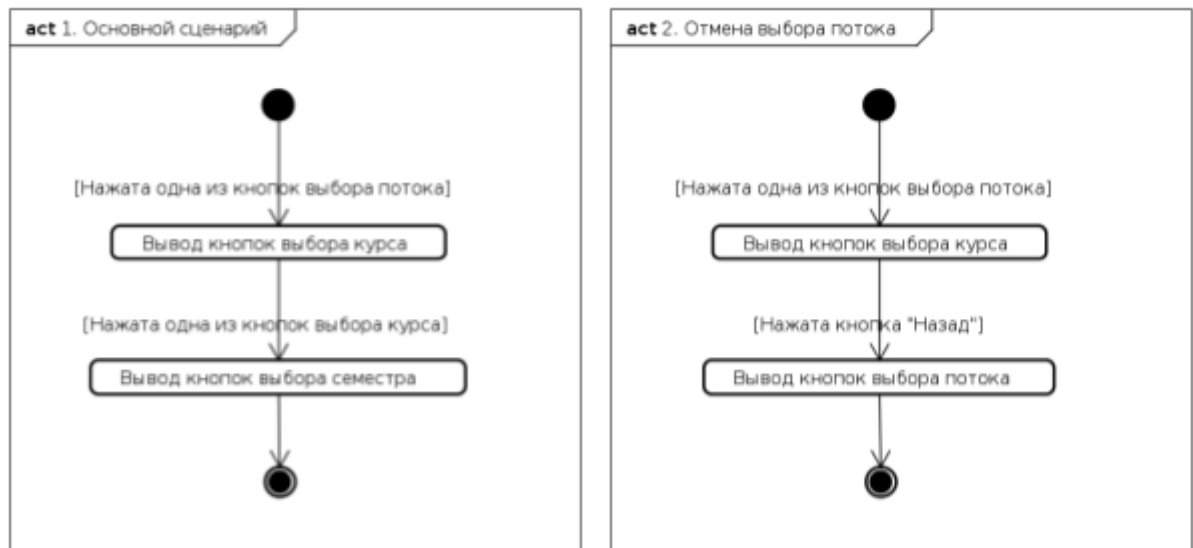


Рис. 5. Основной сценарий при выборе курса и сценарий отмены

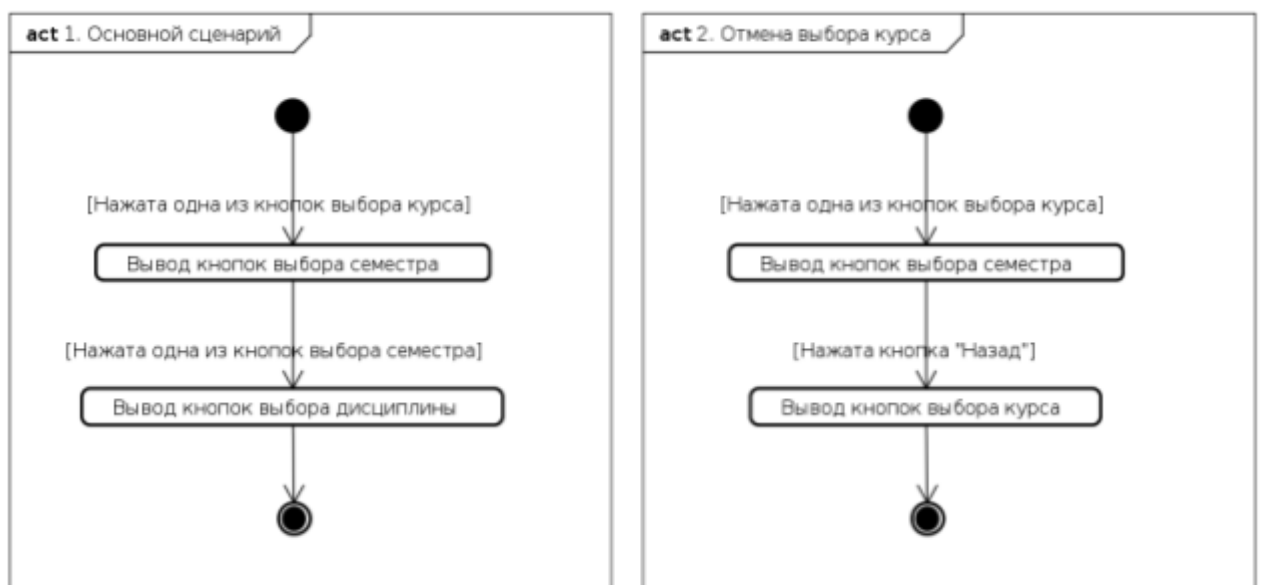


Рис. 6. Основной сценарий при выборе семестра и сценарий отмены

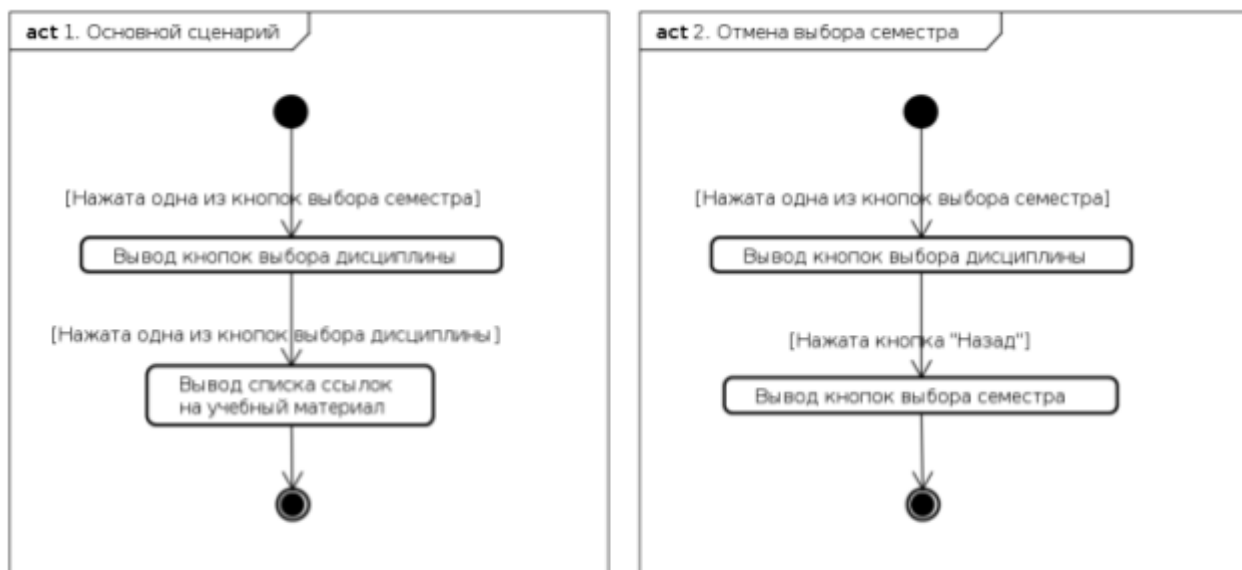


Рис. 7. Основной сценарий при выборе дисциплины и сценарий отмены

3.3 Функциональные требования для роли «Незарегистрированный пользователь»

Use-case и их сценарии для роли «Незарегистрированный пользователь» описаны в *diagrams/1. Use-case модель/UC незарегистрированного пользователя*^[1] (см. Рис. 8).

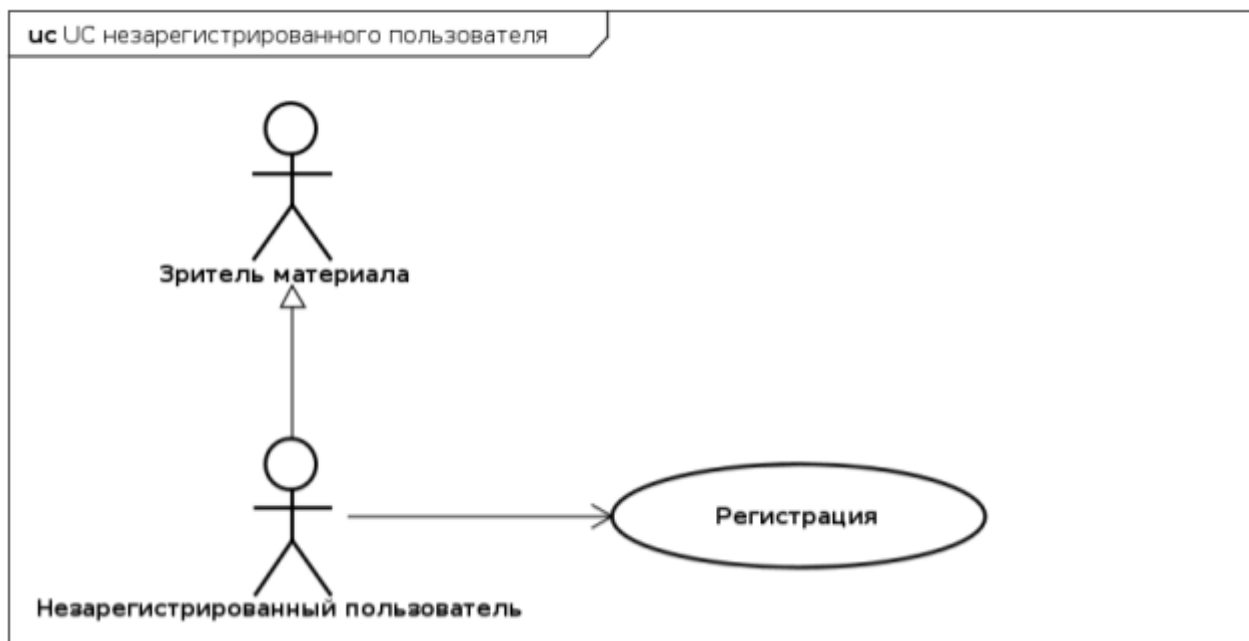


Рис. 8. Use cases Незарегистрированного пользователя

3.3.1 Регистрация

Незарегистрированный пользователь имеет возможность зарегистрироваться для получения доступа к дополнительным функциям системы (см. Рис. 9). Для регистрации необходимо ввести почту НГУ и подтвердить ее при помощи токена из письма.

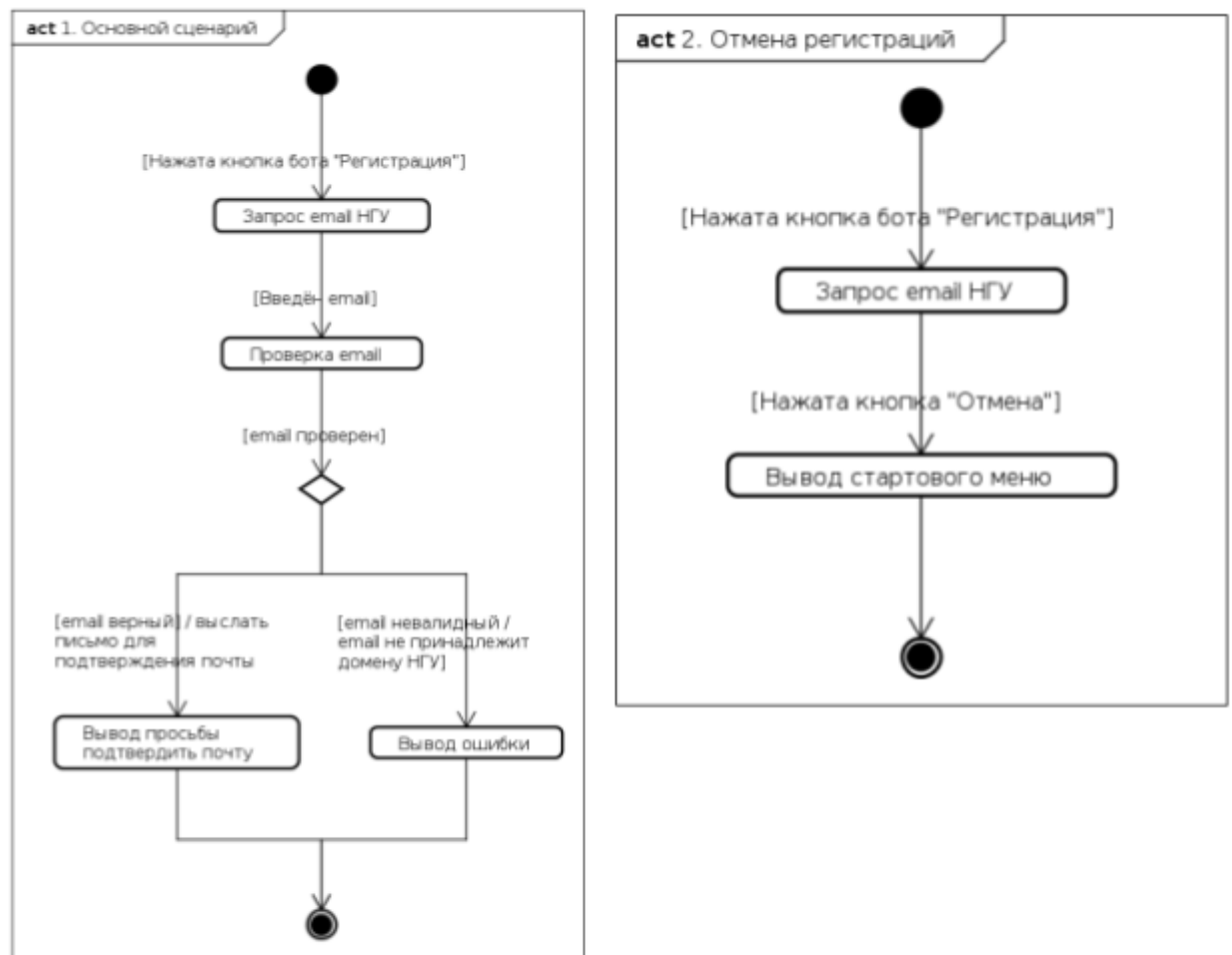


Рис. 9. Основной сценарий регистрации и сценарий отмены

3.4 Функциональные требования для роли «Неактивированный пользователь»

Use-case и их сценарии для роли «Неактивированный пользователь» описаны в *diagrams/1. Use-case модель/UC неактивированного пользователя^[1]* (см. Рис. 10).

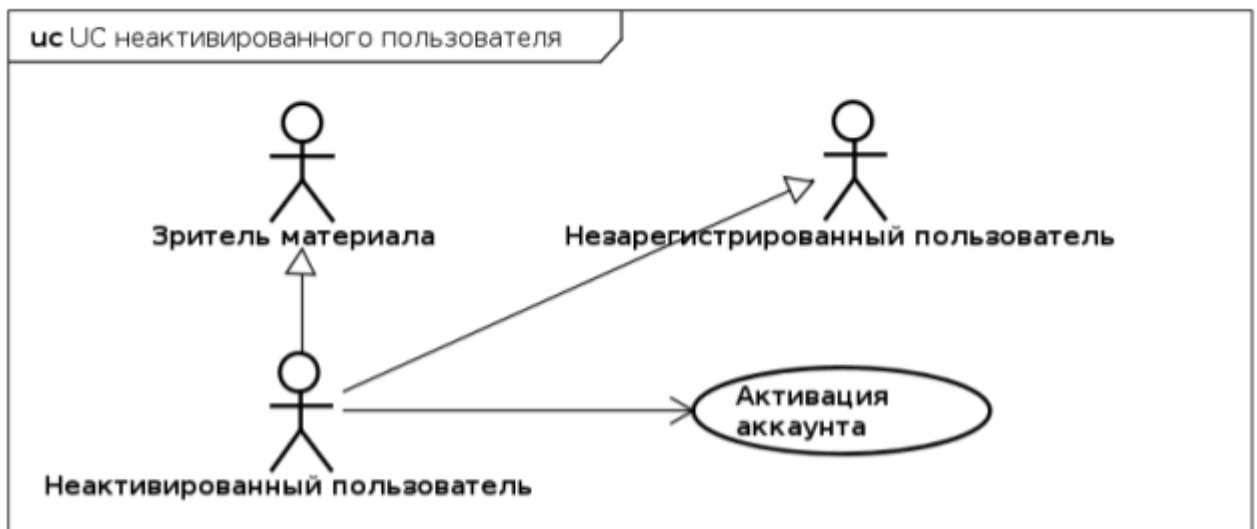


Рис. 10. Use cases Неактивированного пользователя

3.4.1 Активация аккаунта

Неактивированный пользователь может активировать аккаунт при помощи токена, отправленного на ранее введенную почту (см. Рис. 11).



Рис. 11. Основной сценарий активации аккаунта

3.5 Функциональные требования для роли «Зарегистрированный пользователь»

Use-case и их сценарии для роли «Зарегистрированный пользователь» описаны в *diagrams/1. Use-case модель/UC зарегистрированного пользователя^[1]* (см. Рис. 12).

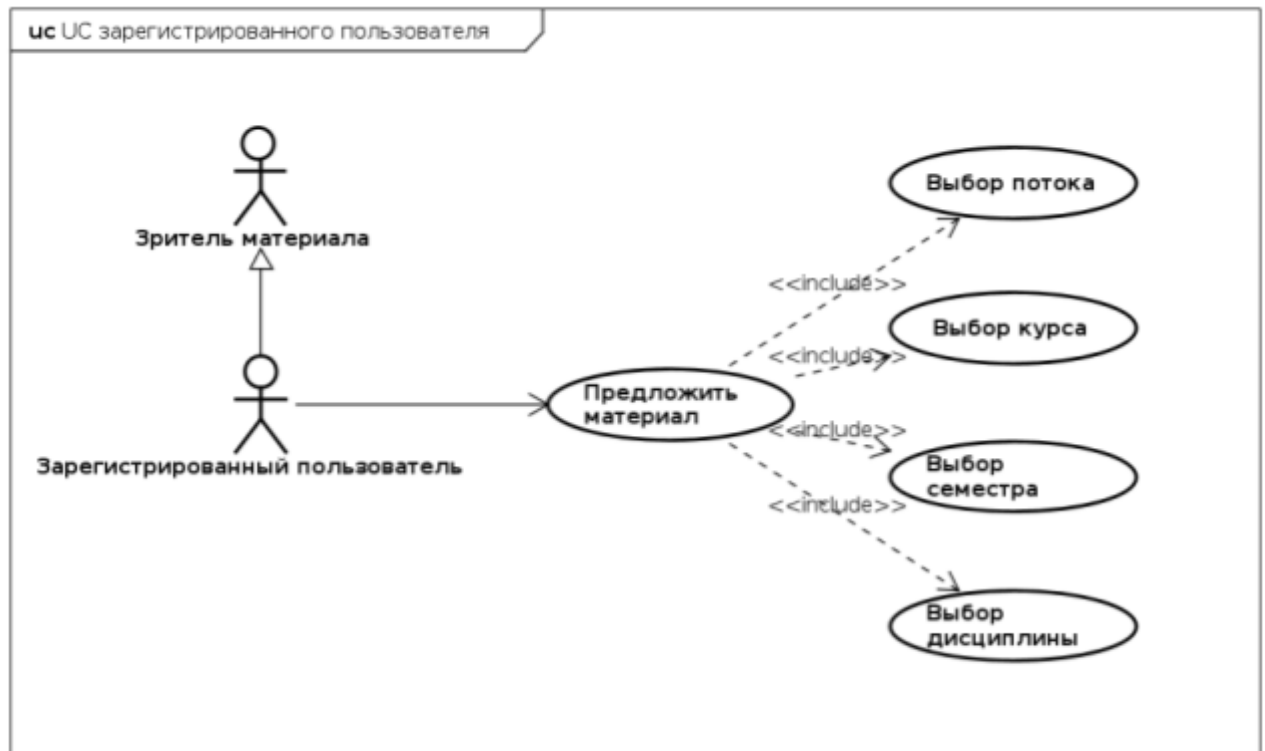


Рис. 12. Use cases Зарегистрированного пользователя

3.5.1 Предложение материала

Зарегистрированный пользователь может предложить новый учебный материал или ресурс для системы. Необходимо последовательно выбирать ту или иную кнопку выбора потока/курса/семестра/дисциплины и загрузить файл пособия (см. Рис 4-7, 13).

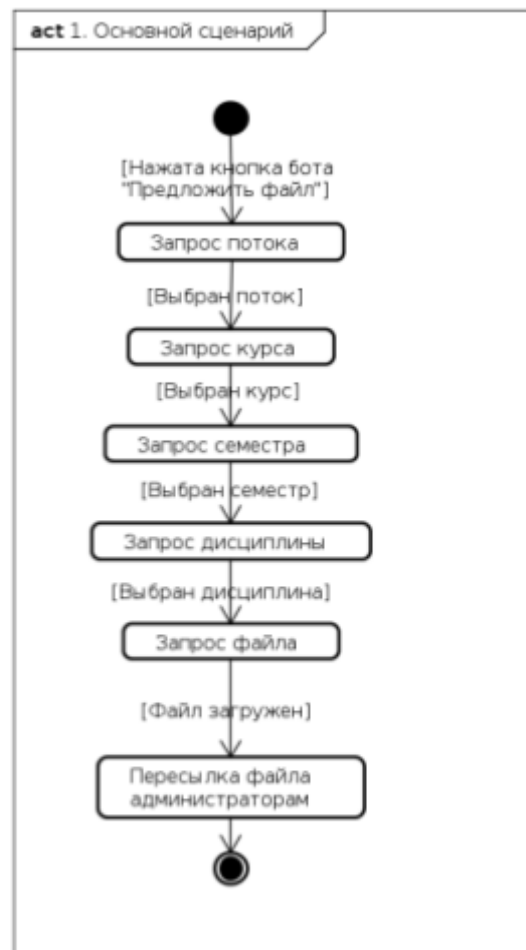


Рис. 13. Основной сценарий предложения материала

3.6 Функциональные требования для роли «Администратор»

Use-case и их сценарии для роли «Администратор» описаны в *diagrams/1. Use-case модель/UC администратора^[1]* (см. Рис. 14).



Рис. 14. Use cases Администратора

3.6.1 Принятие и отклонение материала, отправленного пользователем

Администратор может эффективно управлять материалами, предоставленными пользователями, и обеспечивает процесс принятия или отклонения материалов в зависимости от их соответствия критериям и требованиям ресурса (см. Рис. 15).

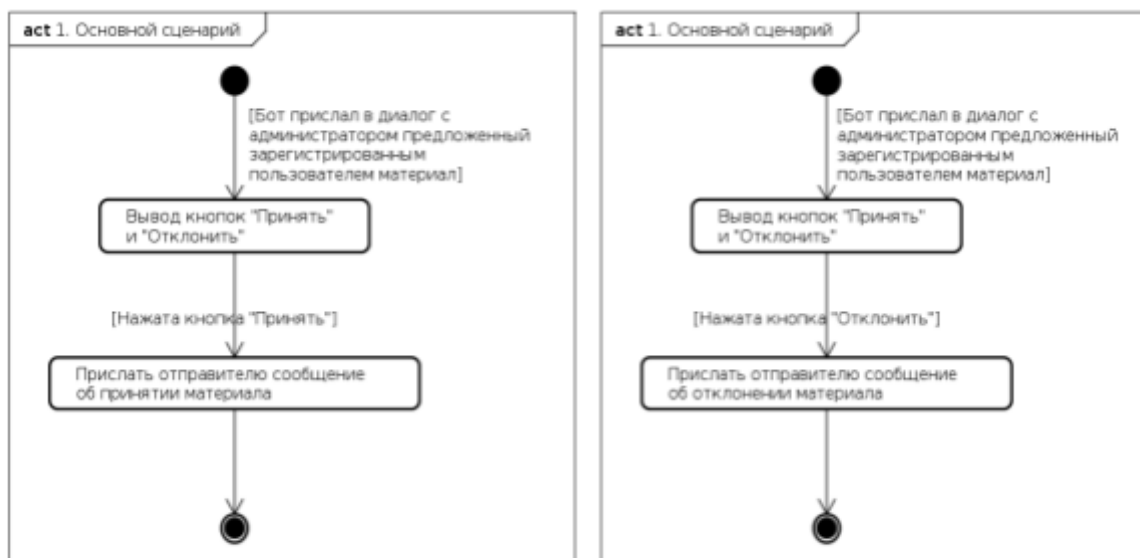


Рис.15. Основные сценарии принятия и отклонения материала пользователя

3.6.2 Добавление материала

Администратору предоставлена возможность добавлять материал на диск через бота. Необходимо указать поток, курс, семестр и дисциплину и прикрепить файл пособия. После подтверждения файл запишется на диск (см. Рис 4-7, 16).



Рис.16. Основной сценарий добавления материала

3.6.3 Удаления материала

Администратору предоставлена возможность удалять материал с диска через бота. Необходимо указать поток, курс, семестр, дисциплину и файл пособия. После подтверждения файл удалится с диска (см. Рис 4-7, 17).



Рис.17. Основной сценарий удаления материала

3.7 Нефункциональные требования

3.7.1 Производительность

Ресурсы (например, вычислительная мощность и хранилище) должны быть масштабируемыми и достаточными для обслуживания пользователей в количестве студентов ФИТ НГУ. Хранилище диска должно иметь возможность хранить учебный материал в объеме 50 ГБ.

3.7.2 Необходимость миграции данных из legacy системы

Данные из legacy системы должны быть корректно и безопасно перенесены в новую систему, сохраняя целостность и доступность. Должна быть выстроена необходимая файловая структура диска для использования ботом.

3.7.3 Использование Google Drive API

Бот должен интегрироваться с API Yandex диска для возможности хранения, получения и обновления учебного материала.

4. Обзор архитектуры

Этот раздел описывает архитектуру системы.

4.1 Компонентная модель системы

[Здесь приводится Component diagram - диаграмма компонентов системы, со связями между компонентами и интерфейсами между ними, а также описание их взаимодействия. Для каждого компонента дается краткое описание его места и предназначения в системе]

4.1.1 Компонент 1

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

4.1.2 Компонент 2

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

4.2 Компоненты сторонних производителей

[Здесь приводится список использованных компонент сторонних производителей, использованных при разработке системы, с указанием их предназначения в системе]

4.3 Схема развертывания приложения

[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]

5. Допущения и ограничения

[Краткое описание допущений, которые подразумевает данный проект, и любых ограничений (например, по бюджету, участникам, требуемому оборудованию, срокам и т.п.), накладываемых на его выполнение.]

Пример: При разработке проекта принято допущение, что число транзакций в единицу времени значительно (более чем в 10 раз) снижается в ночное время, что позволяет в период с 01:00 до 6:00 производить автоматическое обновление программного обеспечения системы, требующее полной перезагрузки и остановки сервиса на период до 5 минут.

6. Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

6.1 Невысокая производительность приложения

Проблема	Производительность приложения экспоненциально деградирует при общем числе пользователей выше 10000 и числе одновременных сессий выше 100.
Ранг	10 (высокий)
Влияние на проект	Невозможность использования системы при числе пользователей более 10000.
Пути решения	Кластеризация веб-сервера и сервера базы данных, а также применение load balancer в точке маршрутизации запроса к веб-серверу.

Лист регистрации изменений

Дата	Версия	Описание	Автор

[В качестве описания версии можно указывать какие изменения/дополнения были сделаны в этой версии по отношению к предыдущей.]

Лист регистрации проверок

Дата	Версия	Описание	Автор

[Здесь описываются результаты проверки документа. Для каждой проверки указывается число, версия документа, описание результатов проверки и имя человека, который делал проверку.]