# 1.1. ЭСКИЗНЫЙ ПРОЕКТ

# 1.1.1 ПРЕЗЕНТАЦИЯ ИДЕИ ПРОЕКТА

## НАЗВАНИЕ ПРОЕКТА

BuyMyPoem - проект, направленный на реализацию взаимодействия авторов литературных произведений с читателями. Он упростит процесс покупки и продажи результатов творческой деятельности, позволит выставлять любым писателям свои работы для просмотра и оценки сообществу, а также поможет им разрабатывать новые проекты с помощью системы антиплагиата, черновиков и других специальных возможностей.

## ЦЕЛЬ И ЗАДАЧИ

Разработать программную систему целью которой является хранение текстовых данных и автоматизация процесса продажи и покупки результатов творческой деятельности, таким образом чтобы любой желающий мог просматривать выгруженные работы в режиме онлайн, взаимодействовать с другими пользователями, организовывая сделки, и загружать собственные работы.

Задачи, которые необходимо решить для достижения поставленной цели:

* грамотно организовать структуру веб-приложения и базы данных
* добавить в веб-приложение функции многопоточности
* разработать систему регистрации, идентификации, аутентификации и авторизации в веб-приложении
* разработать удобный пользовательский интерфейс

## СПИСОК АНАЛОГОВ

* Сервис интернет-рекрумента - headhunter.ru
* Архив хранения текстовых данных, размещаемых пользователями на безвозмездной основе – ficbook.ru
* Сервис проверки текста на уникальность – antiplagiat.ru

## РОЛИ КОМАНДЫ. РАСПРЕДЕЛЕНИЕ ЗАДАЧ ПО ИСПОЛНИТЕЛЯМ.

(на этап проектирования)

Журавлев Н.М. – Анализ предметной области, создание диаграммы прецедентов и диаграммы состояний.  
Хлызова В.Г. – Анализ предметной области, создание эскиза интерфейса, создание диаграммы классов.

# 1.1.2. КАЛЕНДАРНЫЙ ПЛАН ПРОЕКТА

## СПИСОК ОСНОВНЫХ ЗАДАЧ ПО СРОКАМ ВЫПОЛНЕНИЯ

пункт 1.1.1. Презентация идеи проекта: deadline - 01.11.2019

пункт 1.1.2. Календарный план проекта: deadline - 08.11.2019

пункт 1.1.3. State of art: deadline - 08.11.2019

пункт 1.1.4. Анализ предметной области: deadline - 08.11.2019

пункт 1.1.5. Макет программной системы: deadline - 15.11.2019

# 1.1.4. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область подразумевает хранилище текстов с реализацией функции автоматизации процессов обмена результатами творческой деятельности. (Процесс купли-продажи литературных произведений).

## СЛОВАРЬ ПРЕДМЕТНОЙ ОБЛАСТИ

Author(Автор) – пользователь системы, просматривающий и добавляющий работы

Customer(Заказчик) – пользователь системы просматривающий и заказывающий работы.

PaymentResourse(Платежное средство) – абстракция, содержащая данные о платежных ресурсах пользователя.

Order(Заказ) – абстракция, описывающая и реализующая процесс обмена выполненной работы на денежные средства.

Request(Заявка) – абстракция, описывающая требования заказчика к будущей сделки

Message(Сообщение) – абстракция, описывающая реализующая процесс взаимодействия и общения между пользователями

Composition(Произведение) – абстракция, описывающая свойства результата творческой деятельности.

Status(Статус) – Статус произведения (черновик, открытая работа, заказная работа)

Genre(Жанр) – Жанр произведения (Стихотворение, рассказ, текст песни и т.д.)

Tag(Тег) - абстракция, описывающая кодовые слова для краткого и общего описания и поиска произведения.

## ПОЛЬЗОВАТЕЛИ СИСТЕМЫ И РОЛИ

Гость – неавторизованный пользователь, может просматривать авторизированных пользователей, произведения авторов и заявки. Имеет возможность авторизоваться или создать учётную запись.

Авторизованный пользователь:

Заказчик – пользователь, авторизованный в системе. Представляет физическое лицо, которое выступает в качестве покупателя результата творческой деятельности

Автор - пользователь, авторизованный в системе. Представляет физическое лицо, которое выступает в качестве продавца результата творческой деятельности

## ПРЕЦЕДЕНТЫ



СПЕЦИФИКАЦИЯ ПРЕЦЕДЕНТОВ

**1. “Подтвердить выполнение заказа”**

**Предусловие.** Автор откликнулся на заявку, заказчик выбрал автора, автор выполнил работу.

**Актеры:** “Заказчик”

**Основной поток.** Актер просматривает работу выполненную автором и, если его устраивает результат, он нажимает на кнопку подтверждения выполнения.

**Альтернативный поток.** Актера не устраивает результат выполнения работы после предварительного просмотра. Он нажимает на кнопку отправки заказа на редактирование, после чего ему предлагается написать комментарий к работе.

**Постусловие.** Актер начинает выполнение прецедента “Оплатить заказ”.

**2. “Оценить работу”**

**Предусловие:** Актер просматривает опубликованную работу.

**Актеры:** “Автор”, “Заказчик”

**Основной поток:** Актер нажимает на кнопку “Понравилось”, количество отметок “Понравилось” увеличивается на 1; или же актер нажимает на кнопку “Не понравилось”, количество отметок “Не понравилось” увеличивается на 1.

**Альтернативный поток:** Источник данных временно не доступен, Актёр получает уведомление с сообщением о возникших проблемах.

**Постусловие:** На экран выводится список всех опубликованных работ.

3.**“Посмотреть список заявок”**

**Предусловие: -**

**Актеры:** “Автор”

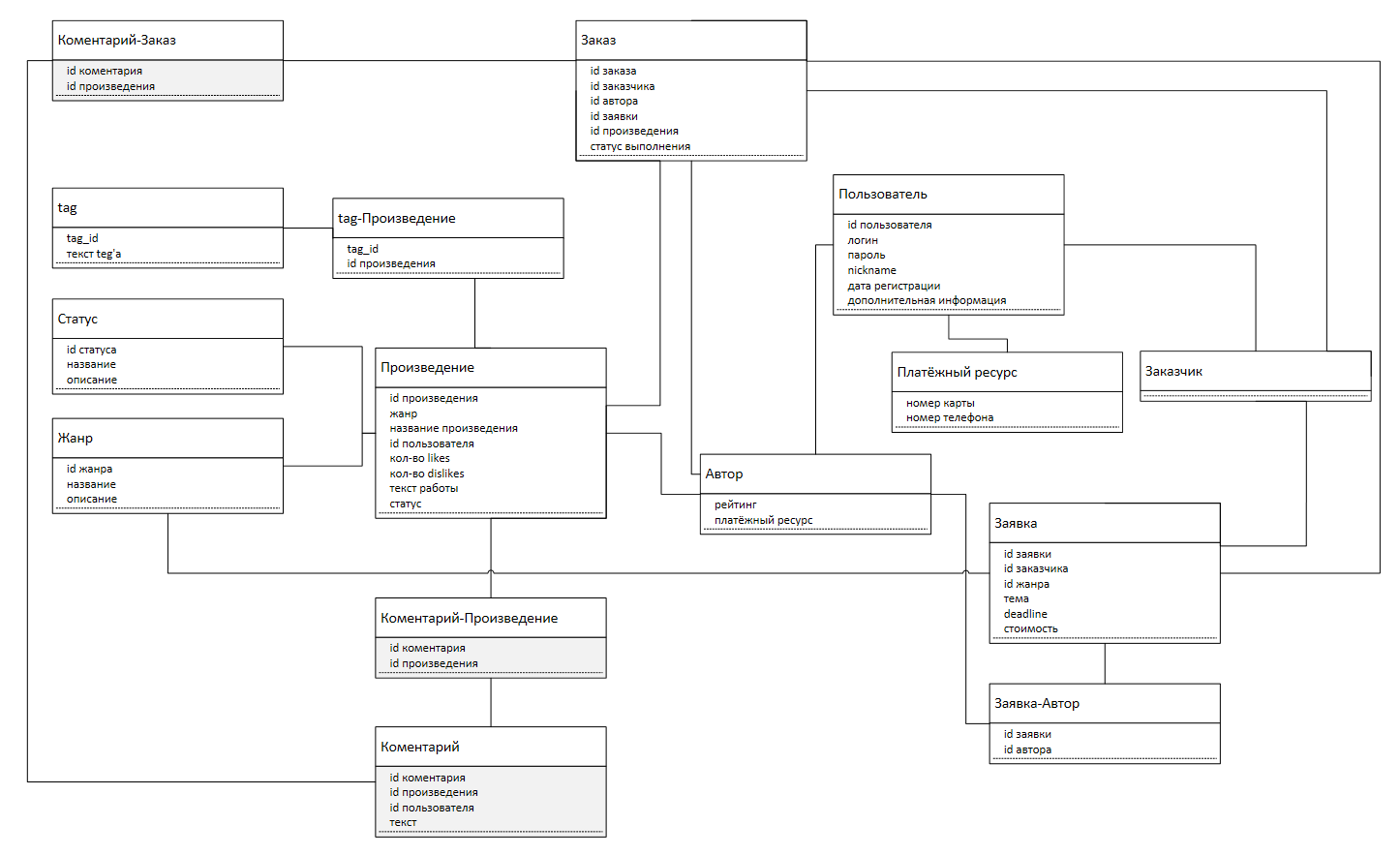
**Основной поток:** Актёр с главной страницы переходит по кнопке “Список заявок” на страницу со списком актуальных заявок от Заказчиков.

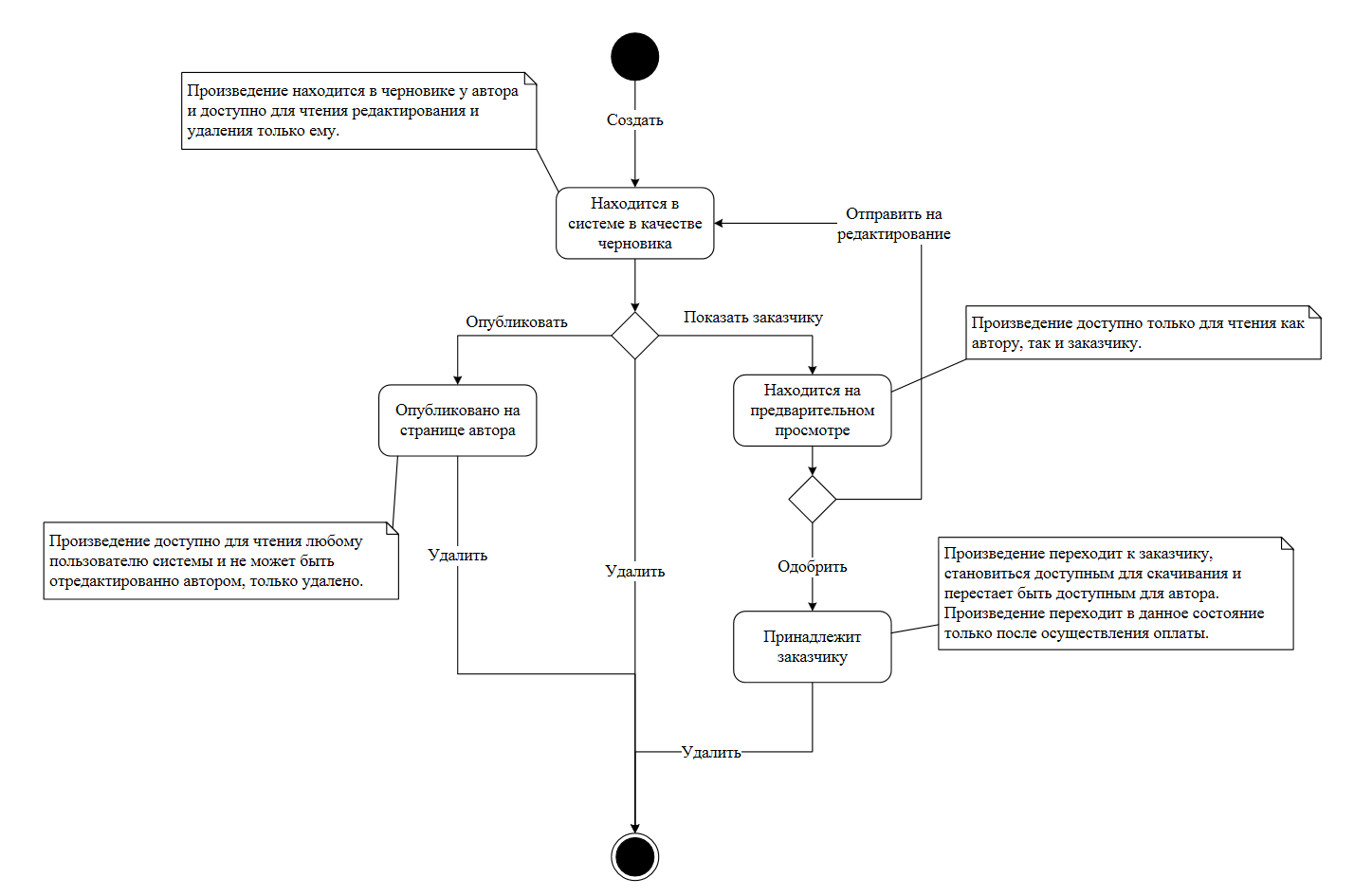
**Альтернативный поток:** Источник данных временно не доступен, Актёр получает уведомление с сообщением о возникших проблемах.

**Постусловие:** Актёр пролистывает список всех заявок и выбирает понравившуюся заявку либо осуществляет поиск по заявкам и ищет подходящую.

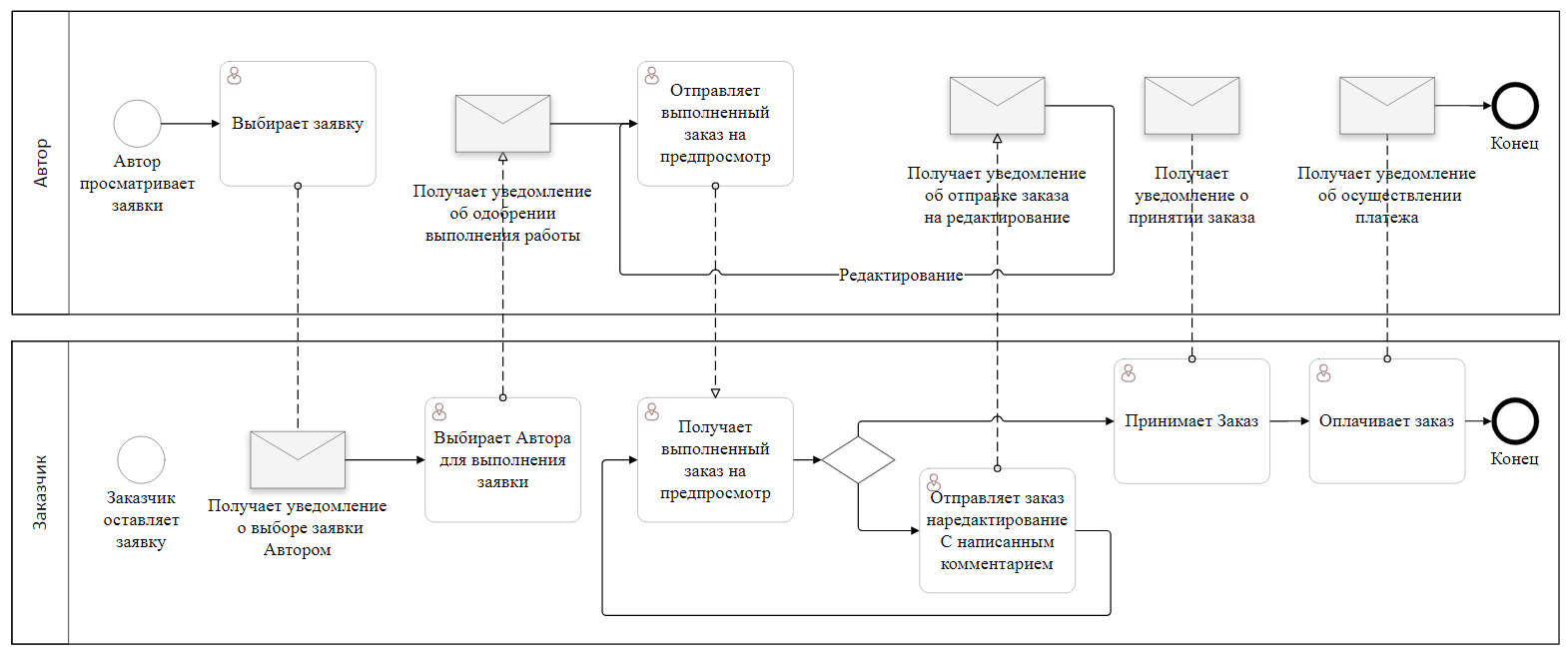
## ОПИСАНИЕ ДОКУМЕНТОВ/СУЩНОСТЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ

1. Концептуальная диаграмма классов (UML Class Diagram)



2. Диаграмма состояний произведения 

## АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ



## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ

* Система должна обеспечивать возможность регистрации, авторизации, аутентификации и идентификации пользователей.
* Система должна обеспечивать хранение текстовых данных и управление ими.
* Система должна обеспечивать организацию бизнес-логики для процесса купли и продажи результата творческой деятельности.
* Система должна включать в себя функцию проверки текста на уникальность.

## НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ

* Система должна быть реализована в виде web-приложения
* Система должна предусматривать базовую защиту от SQL-инъекций
* Система должна иметь простой и понятный пользователю интерфейс
* Система должна предусматривать валидацию данных
* Система должна обеспечивать целостность данных и оптимизацию управления ими.

# 1.1.5. МАКЕТ ПРОГРАММНОЙ СИСТЕМЫ