

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук
Кафедра Программирования и информационных технологий

Курсовой проект

*Веб-приложение для публикации и просмотра новостей
“Публичный корпоративный блог”*

09.03.04 Программная инженерия

Обучающийся _____ *Бородин А.О., 3 курс*

Обучающийся _____ *Буйлов Н.О., 3 курс*

Обучающийся _____ *Свиридов М.А., 3 курс*

Руководитель _____ *А.В. Нужных, преподаватель*

Воронеж, 2020

Введение

Блоги бывают корпоративными и персональными. В этом курсовом проекте мы рассмотрим процесс производства корпоративного блога. Под корпоративным блогом понимается блог, который ведется группой людей или организацией.

Корпоративный блог может использоваться исключительно внутри организации — такой блог называют внутрикорпоративным. Другая разновидность блога — это публичный блог компании. в отличие от внутрикорпоративного, он нацелен на общение с широким кругом клиентов и деловой средой. в нашей книге речь пойдет как раз о публичном корпоративном блоге.

Корпоративный блог — это не просто новостная лента компании. Сама суть блога подразумевает диалог между компанией и читателями.

Постановка задачи

Цель курсовой работы – работая в команде, разработать веб-приложение для ведения публичного корпоративного блога, распространения новостей и уведомления о новых мероприятиях компании.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Анализ предметной области
2. Изучение существующих решений
3. Анализ требований к системе
4. Анализ продуктовых сценариев
5. Проектирование системы
6. Выбор методов и средств разработки
7. Реализация функциональности веб-приложения
8. Реализация пользовательского интерфейса
9. Тестирования веб-приложения

1. Аналитическая часть

1.1. Анализ предметной области

Зачем компания заводит корпоративный блог? Блог заводят ради двух основных целей — информирования и продвижения.

Один блог может выполнять несколько функций одновременно. Рассмотрим самые популярные из них.

1.1.1. Информирование.

Блог заводят, чтобы выносить наружу процессы, которые происходят внутри компании. Функции информирования:

Новостная. Людям, заинтересованным в деятельности компании, нужно сообщать о новостях и событиях, новых продуктах и возможностях.

Поддержка и обучение. Блог помогает поддерживать связь с клиентами и рассказывать о том, как пользоваться продуктом компании.

Социальная. В больших компаниях, состоящих из множества отделов, сотрудники не всегда знают, что происходит в соседних отделах. В этом случае, корпоративный блог выступает как площадка информирования и социализации.

1.1.2. Продвижение.

Под продвижением можно понять действия, направленные на увеличение узнаваемости компании или продукта, повышение рыночной доли и привлечение новых клиентов.

Увеличение трафика. С помощью экспертных статей, публикуемых в блоге, компания привлекает больше трафика, который способствует увеличению продаж.

Популяризация бренда. Блог способствует рассказать о компании большому количеству людей, не тратясь на рекламу.

Привлечение целевой аудитории. Блог позволяет привлечь внимание разных типов целевой аудитории: это потенциальные клиенты (которые ещё не задумывались о покупке продукта), будущие клиенты (которые раздумывают насчет покупки) и текущие клиенты (которые уже купили продукт).

Развитие индустрии. Публикуемые в блоге статьи экспертов по профессиональной тематике дают индустрии новые знания и двигают ее вперед.

1.2. Анализ существующих решений

1.2.1. blogs.cisco.com

Корпоративный блог, написанный работниками компании, с огромным количеством информации и с удобнейшим фильтром для поиска нужной темы или статьи.

1.2.2. blog.dataart.com

Блог международной сети компаний, которые проектируют, разрабатывают, модернизируют и поддерживают IT-решения.

1.2.3. habr.com

Веб-сайт в формате системы тематических коллективных блогов (именуемых хабами) с элементами новостного сайта, созданный для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом.

1.2.4. vc.ru

Крупнейшая в рунете платформа для предпринимателей и высококвалифицированных специалистов малых, средних и крупных компаний.

1.3. Анализ требований

Анализ существующих решений показал, что приложение должно позволять работникам компании – авторам, публиковать статьи, редактировать их и по необходимости удалять. Также автор должен иметь возможность проводить модерацию комментариев под своими статьями и получать статистику в личном кабинете, связанную со написанными им статьями.

Для обычного пользователя приложение должно обеспечивать свободный просмотр выложенных авторами статей, комментирование их, просмотр информации о компании и возможность связаться с компанией, используя E-Mail. При просмотре статей, пользователь должен иметь возможность фильтровать список всех статей.

В системе должны быть предусмотрены три группы пользователей:

- Администратор
- Автор
- Гость

Гость обладает следующими функциями:

- Просмотр постов, выложенных авторами
- Возможность подписаться на новостную рассылку о новых статьях
- Возможность оставлять комментарии к постам
- Просмотр информации о компании
- Возможность связи с компанией путём E-Mail

Администратор, помимо функций гостя и автора, обладает следующими функциями:

- Удаление и добавление новых авторов
- Удаление любых постов
- Просмотр статистики системы
- Сброс пароля для авторов

На диаграмме вариантов использования (приложение Б №1, №2, №3, №4) отражено, какие функции должна предоставлять система каждому виду пользователей.

Таким образом, в разрабатываемой системе можно выделить следующие классы:

- Статья (Post)
- Автор (Author)
- Комментарий (Comment)
- Категория (Category)
- Подписчик (Mailing Member)

Система должна хранить об авторе следующую информацию:

- Фамилию и имя (опционально)
- Логин
- E-Mail
- Пароль
- Краткую информацию (опционально)
- Список опубликованных статей
- Фотографию (опционально)

Система должна хранить о статье следующую информацию:

- Заголовок
- Текст статьи
- Автор
- Список категорий, к которым эта статья относится
- Картинка-превью
- Дата опубликования
- Дата изменения

Система должна хранить о комментарии следующую информацию:

- Имя отправителя (опционально)
- Автор (опционально)
- Текст сообщения
- Дата опубликования
- Статья
- Комментарий-родитель

Диаграмма классов, описывающая данную систему, приведена в Приложении В. Возможные отношения между экземплярами классов приведены в диаграмме объектов (Приложение Г), а их взаимодействие и возможные сценарии использования на диаграмме взаимодействия (Приложение Д). Временные особенности взаимодействия между объектами системы (автором и гостем) и самой системой отображены на диаграмме последовательностей (Приложения Е №1, №2, №3).

Алгоритм действий системы при реализации основных сценариев показан на диаграмме активности (Приложение Ж), а процесс изменений состояний системы при исполнении этих сценариев отражен на диаграмме состояний (Приложение И).

Система должна быть развернута на сервере, иметь базу данных и клиентскую часть (Приложение К).

1.4. Анализ продуктовых сценариев

В анализе продуктовых сценариев участвуют следующие пользовательские сценарии, главными целями которых являются:

- Просмотр гостем статьи
- Подписка гостем на рассылку
- Добавление автором новой статьи

Скриншот воронок из аналитики Яндекс метрики представлен в Приложении Л.

1.4.1. Просмотр гостем статьи

Для достижения поставленной цели пользователь должен:

- Перейти на главную страницу
- По желанию отфильтровать список постов с помощью категорий
- Выбрать подходящую статью
- Перейти на страницу с содержанием статьи



Переход на главную страницу. Стартовая точка сценария. На главной странице находится список статей, из которых пользователь может выбрать любую, с которой продолжится сценарий.

Фильтр списка статей. Пользователь должен иметь возможность каким-либо образом фильтровать список статей по категориям.

Выбор статьи. Выбор статьи пользователем может основываться на кратком описании статьи или на картинке-превью.

Переход на страницу статьи. После выбора статьи, пользователь переходит по ссылке, ведущей на страницу с полным содержанием статьи.

1.4.2. Подписка гостей на рассылку

Для достижения поставленной цели пользователь должен:

- Перейти на главную страницу
- Выбрать подходящую статью
- Перейти на страницу с содержанием статьи

- Успешно заполнить форму подписки справа статьи или снизу страницы
- Нажать кнопку подписаться



Переход на главную страницу. Стартовая точка сценария. На главной странице находится список статей, из которых пользователь может выбрать любую, с которой продолжится сценарий.

Выбор статьи. Выбор статьи пользователем может основываться на кратком описании статьи или на картинке-превью.

Переход на страницу статьи. После выбора статьи, пользователь переходит по ссылке, ведущей на страницу с полным содержанием статьи.

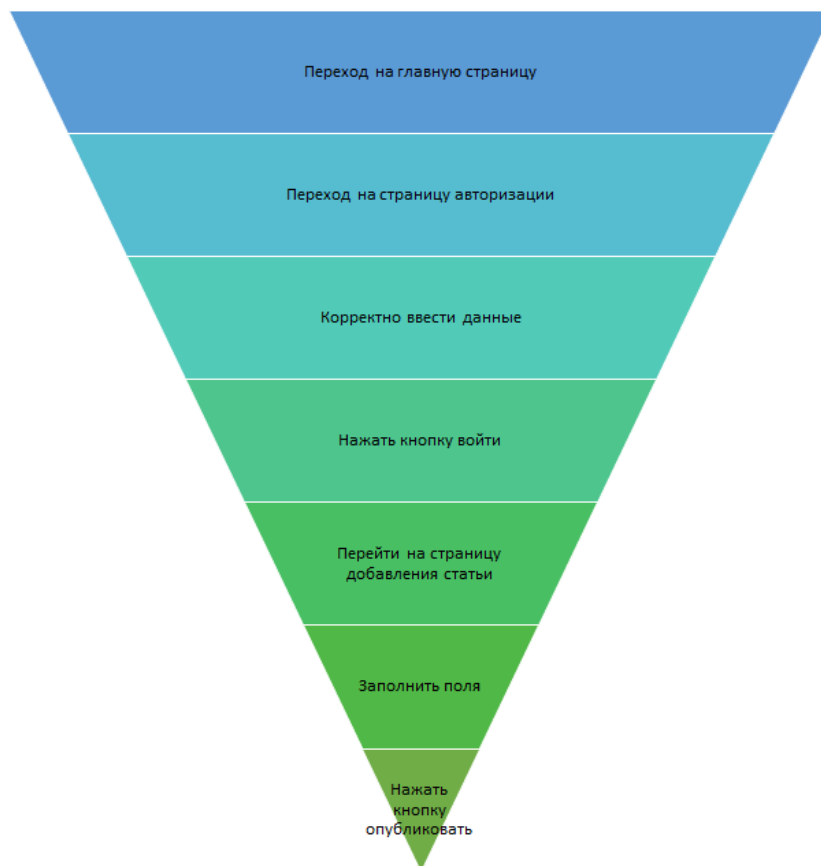
Заполнение формы подписки. На странице с содержанием статьи пользователю предоставляется две формы для заполнения, которые состоят из текстового поля, в которое нужно ввести E-Mail адрес пользователя.

Нажать кнопку подписаться. Подтвердить подписку на рассылку новостей нужно нажатием на кнопку “Подписаться”, которая находится рядом с текстовым полем.

1.4.3. Добавление автором новой статьи

Для достижения поставленной цели пользователь должен:

- Перейти на главную страницу
- Перейти на страницу авторизации
- Корректно ввести данные
- Нажать кнопку войти
- Перейти на страницу добавления новой статьи
- Ввести данные статьи и подтвердить добавление



Переход на главную страницу. Стартовая точка сценария. На главной странице находится ссылка на страницу авторизации для автора.

Переход на страницу авторизации. Для входа в личный кабинет автору нужно авторизироваться в системе.

Корректно ввести данных. Для успешной авторизации, автору нужно корректно ввести логин или E-Mail и пароль.

Нажать кнопку войти. Для входа в личный кабинет после ввода данных пользователю необходимо нажать кнопку “Войти”. После нажатия пользователь попадает в личный кабинет.

Переход на страницу добавления статьи. Для опубликования новой статьи пользователю необходимо перейти на страницу добавления новой статьи. Для этого пользователю нужно нажать на специальную кнопку.

Заполнить поля. На странице с добавлением новой статьи пользователю предоставляется форма для заполнения, которая состоит из текстового поля для заголовка статьи, текстового поля для ввода текста статьи, списка для выбора категорий, поля для загрузки картинки-превью.

Нажать кнопку опубликовать. Подтвердить опубликование статьи нужно нажатием на кнопку “Опубликовать”, которая находится после всех полей формы для добавления статьи.

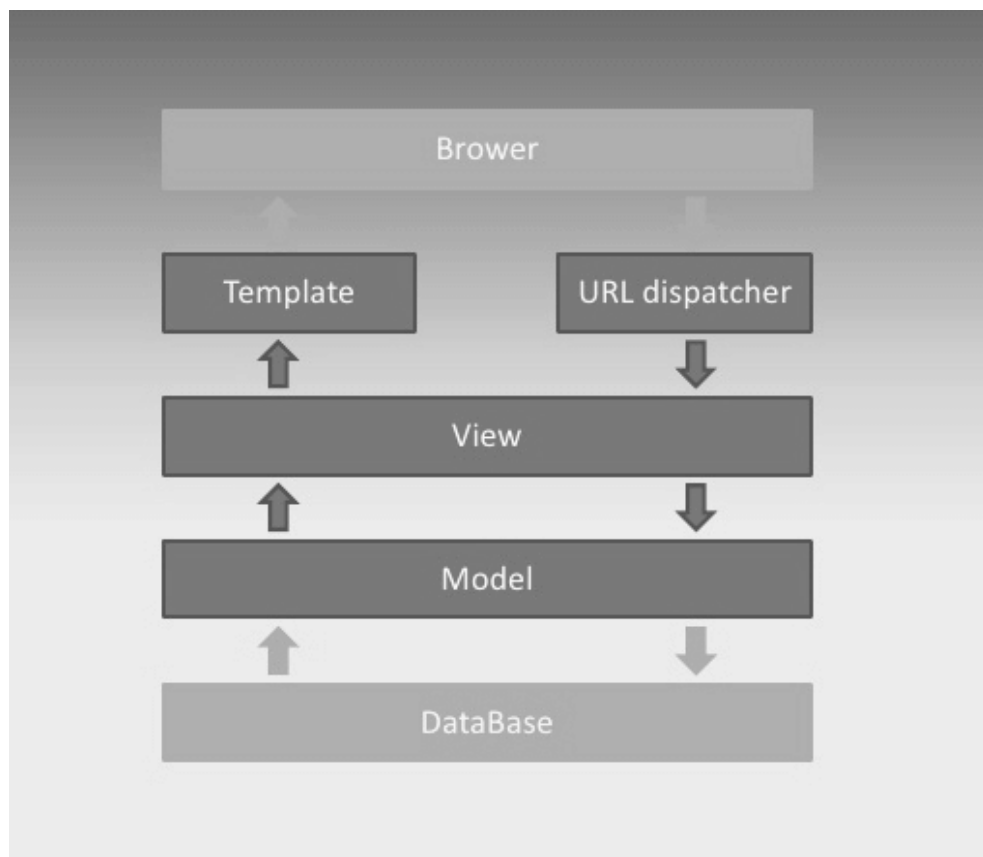
1.5. Выбор методов и средств разработки

Архитектура проекта: при проектировании приложения был использован паттерн проектирования MVT.

Паттерн MVT (Model-View-Template) является модификацией MVC. Основное различие между этими двумя шаблонами состоит в том, что Django сам заботится о части контроллера (программный код, который контролирует взаимодействие между моделью и представлением). Шаблон

представляет собой файл HTML, смешанный с языком шаблонов.

Схематично паттерн можно определить следующим образом:



Основные элементы паттерна:

URL dispatcher: при получении запроса на основании запрошенного адреса URL определяет, какой ресурс должен обрабатывать данный запрос.

View: получает запрос, обрабатывает его и отправляет в ответ пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны. В архитектуре MVC этому компоненту соответствуют контроллеры (но не представления).

Model: описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных.

Template: представляет логику представления в виде сгенерированной разметки html. В MVC этому компоненту соответствует View, то есть представления.

Когда к приложению приходит запрос, то URL dispatcher определяет, с каким ресурсом сопоставляется данный запрос и передает этот запрос выбранному ресурсу. Ресурс фактически представляет функцию или View, который получает запрос и определенным образом обрабатывает его. В процессе обработки View может обращаться к моделям и базе данных, получать из нее данные, или, наоборот, сохранять в нее данные. Результат обработки запроса отправляется обратно, и этот результат пользователь видит в своем браузере. Как правило, результат обработки запроса представляет сгенерированный html-код, для генерации которого применяются шаблоны (Template).

Выбор данного шаблона проектирования был обусловлен фреймворком Django, который используется для разработки системы. Такая архитектура позволяет Django успешно решать разные задачи. Выбранный шаблон проектирования позволяет четко разделить логику приложения, что облегчит поддержку и тестирование кода.

Диаграмма IDEF0

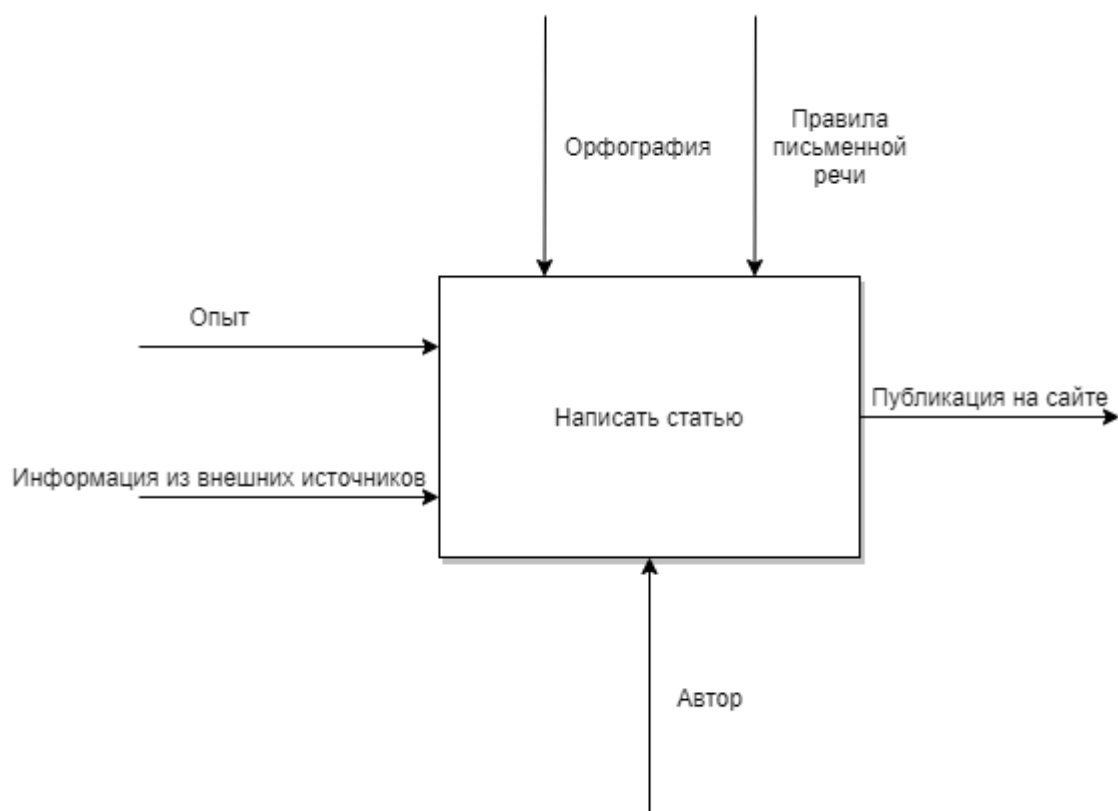


Диаграмма вариантов использования. Администратор.

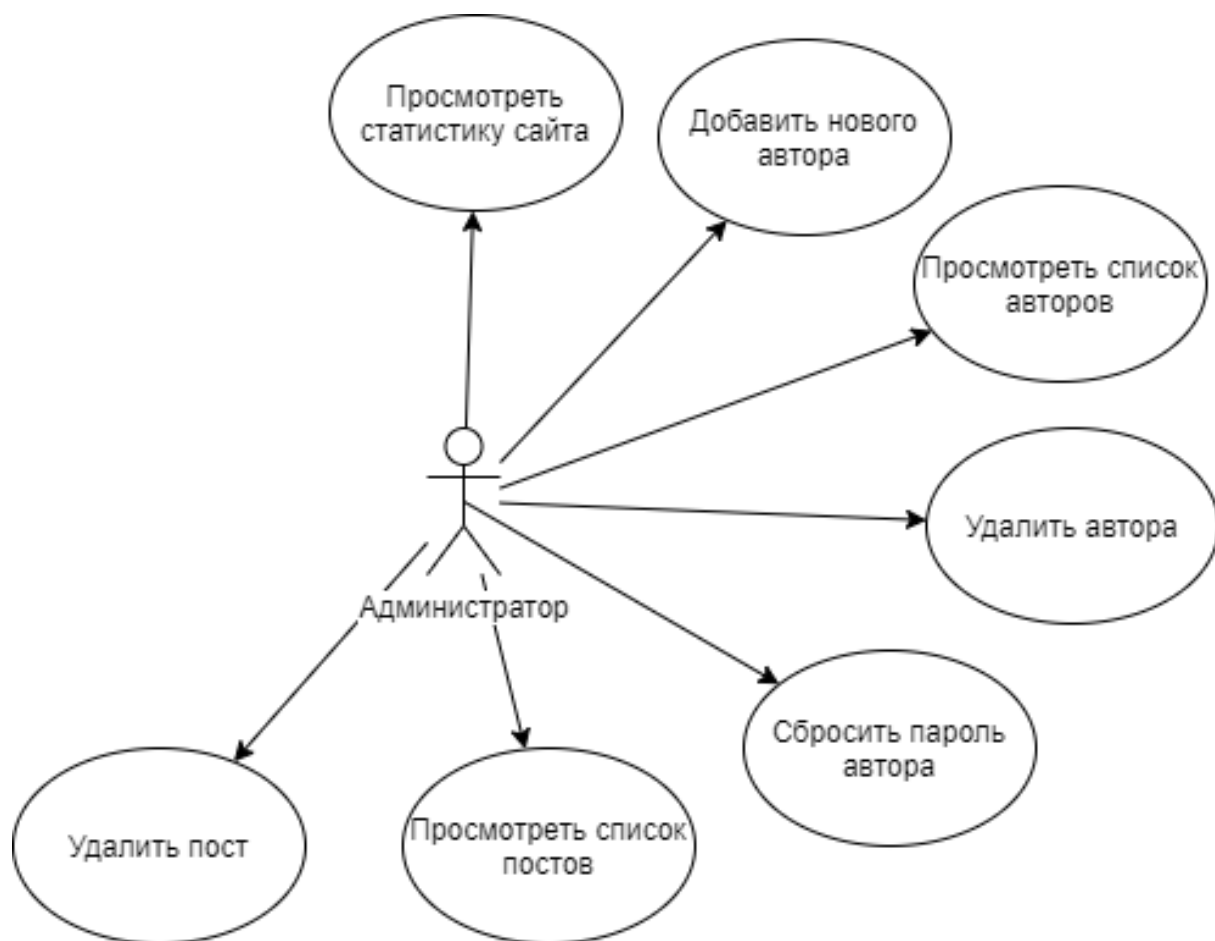


Диаграмма вариантов использования. Автор и гость.

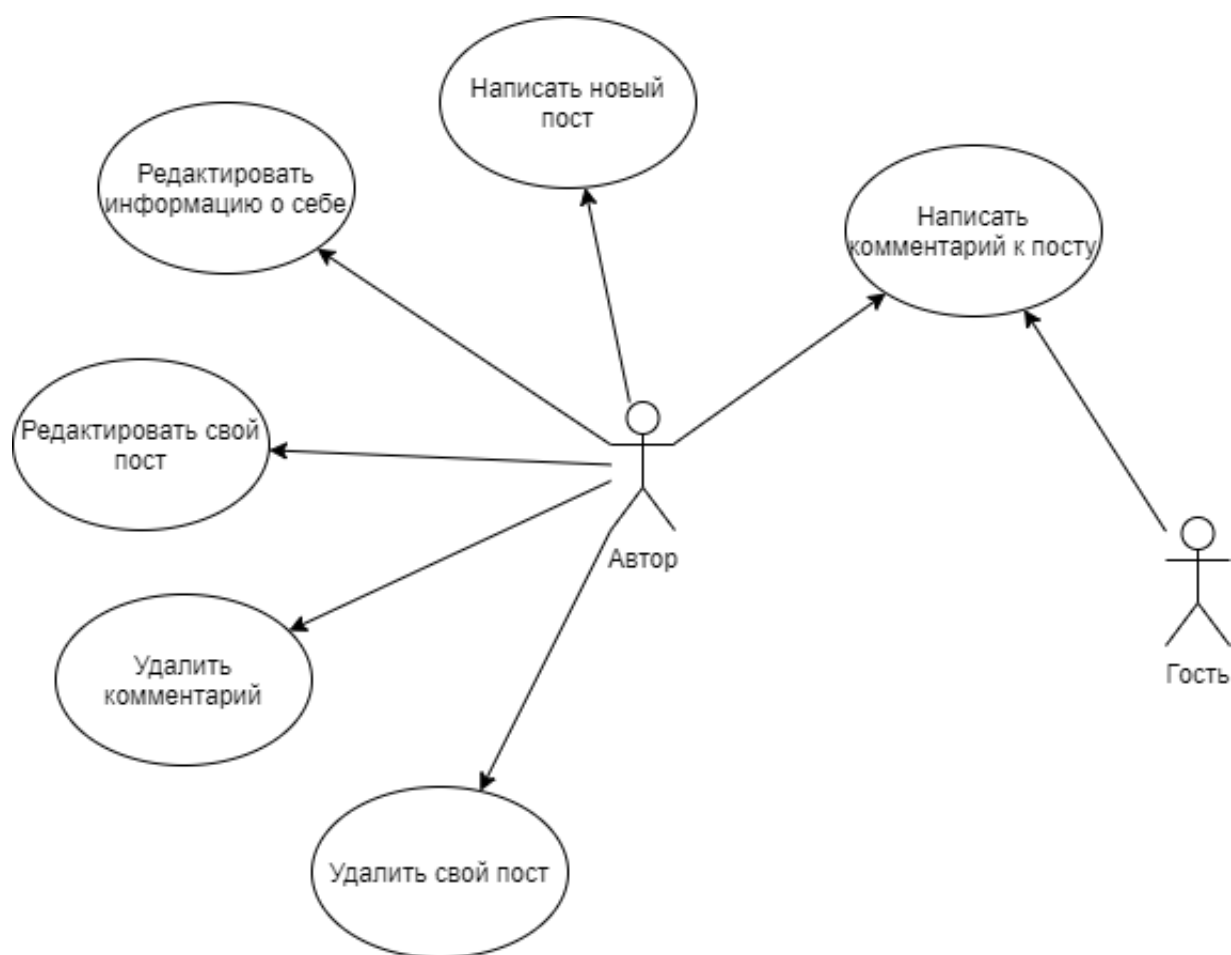


Диаграмма вариантов использования. Авторизация
пользователей.

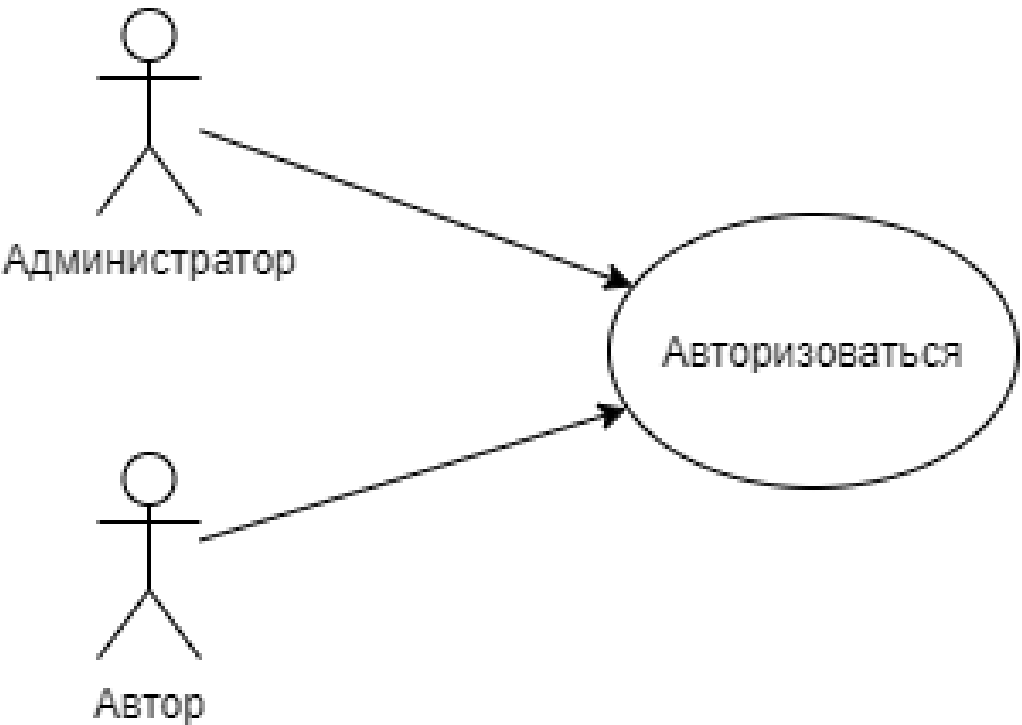
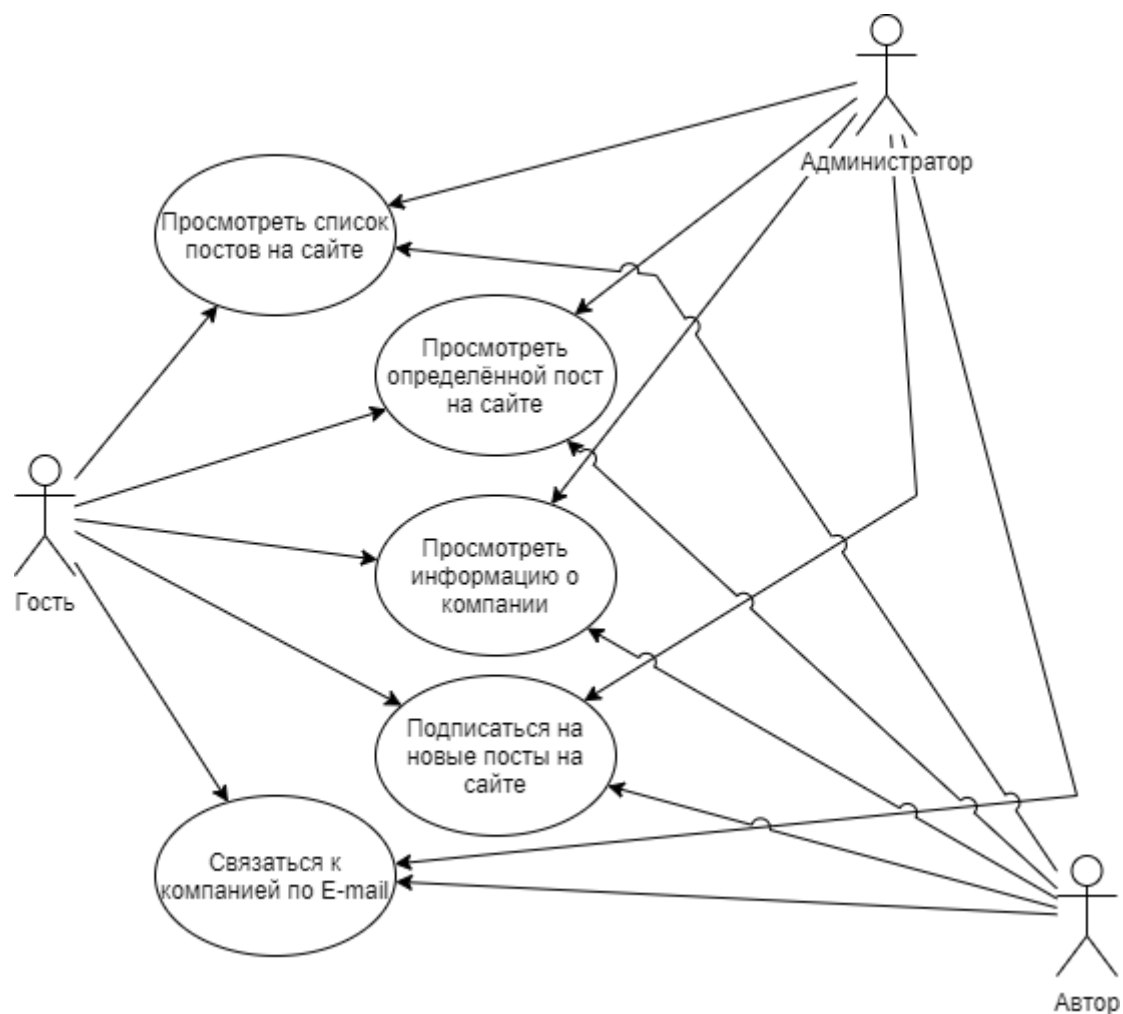


Диаграмма вариантов использования. Общие возможности пользователей.



ПРИЛОЖЕНИЕ В

Диаграмма классов

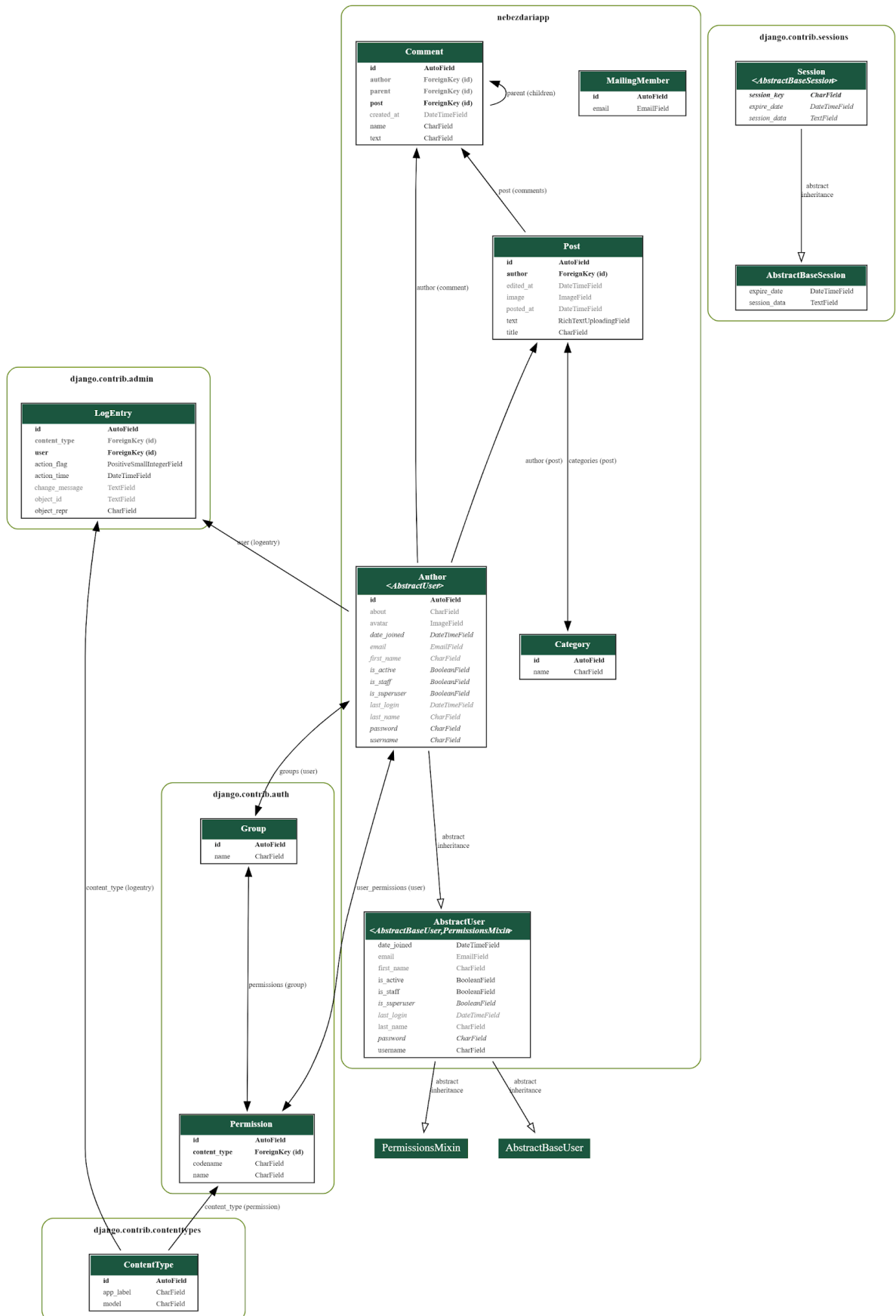
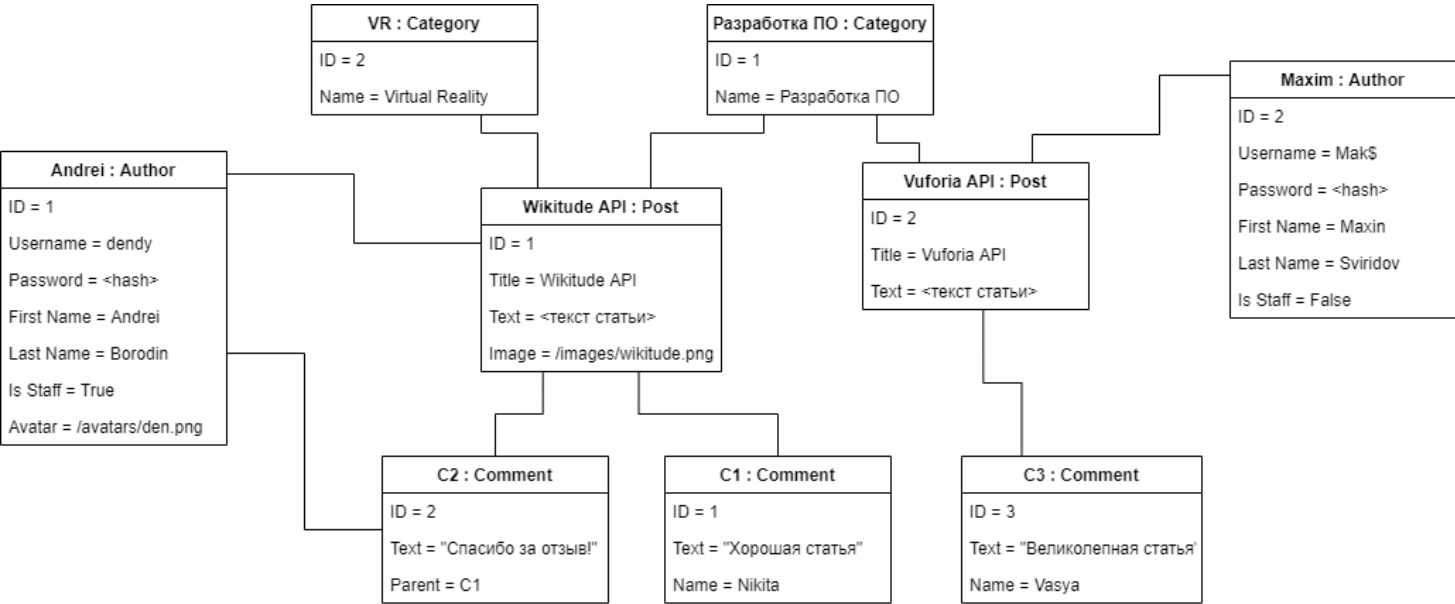


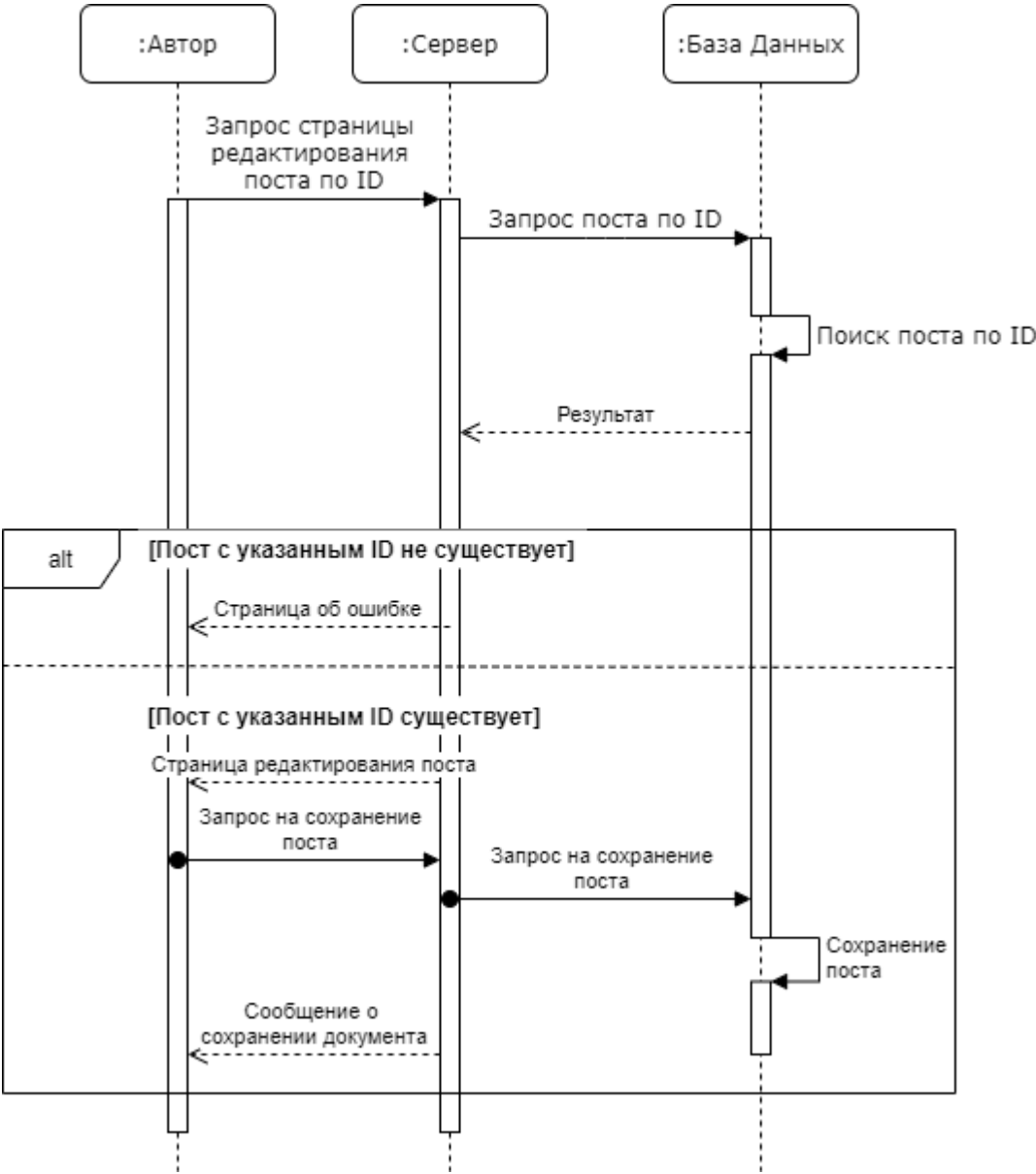
Диаграмма объектов



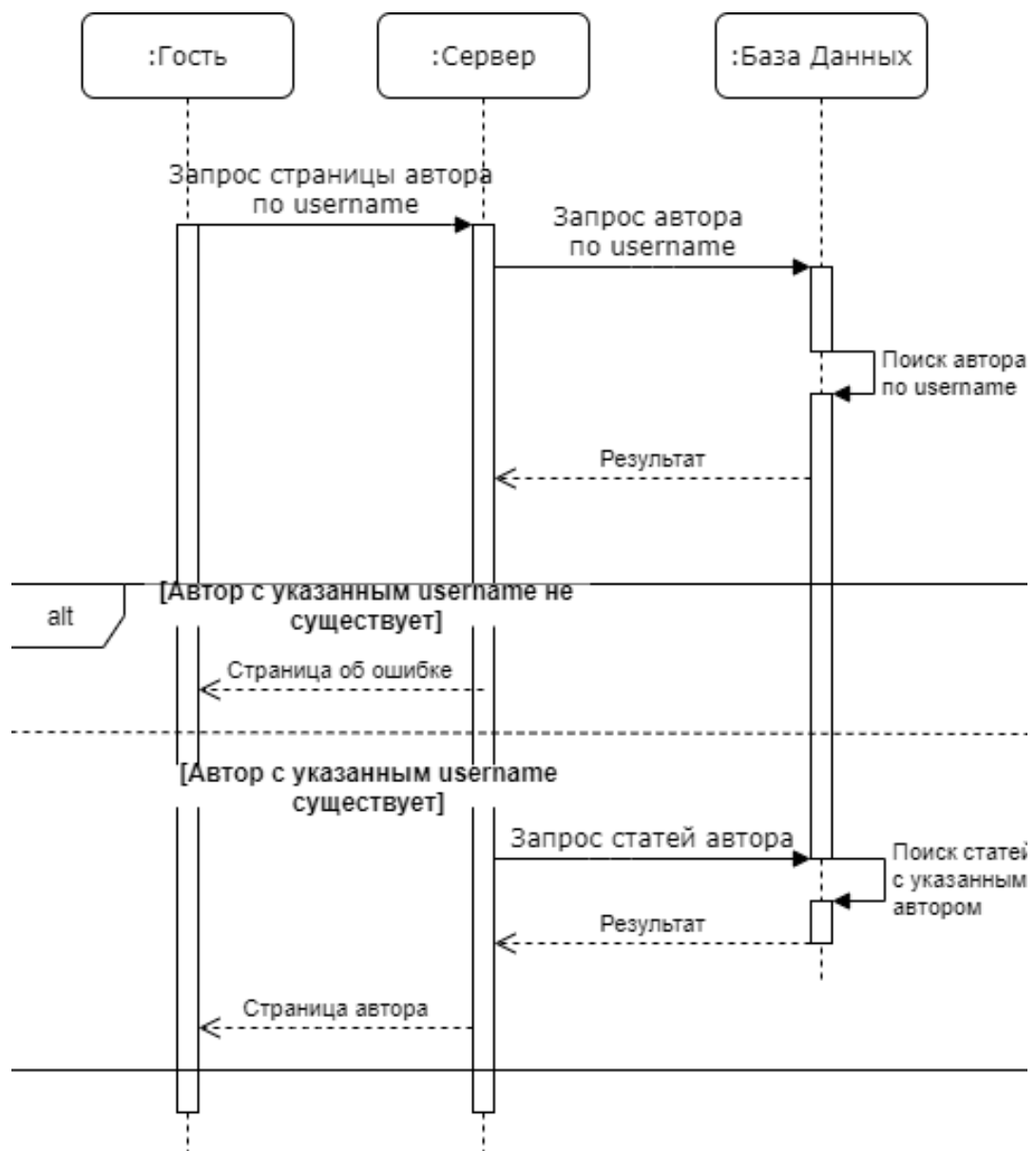
**Диаграмма взаимодействия. Добавление администратором
нового автора.**



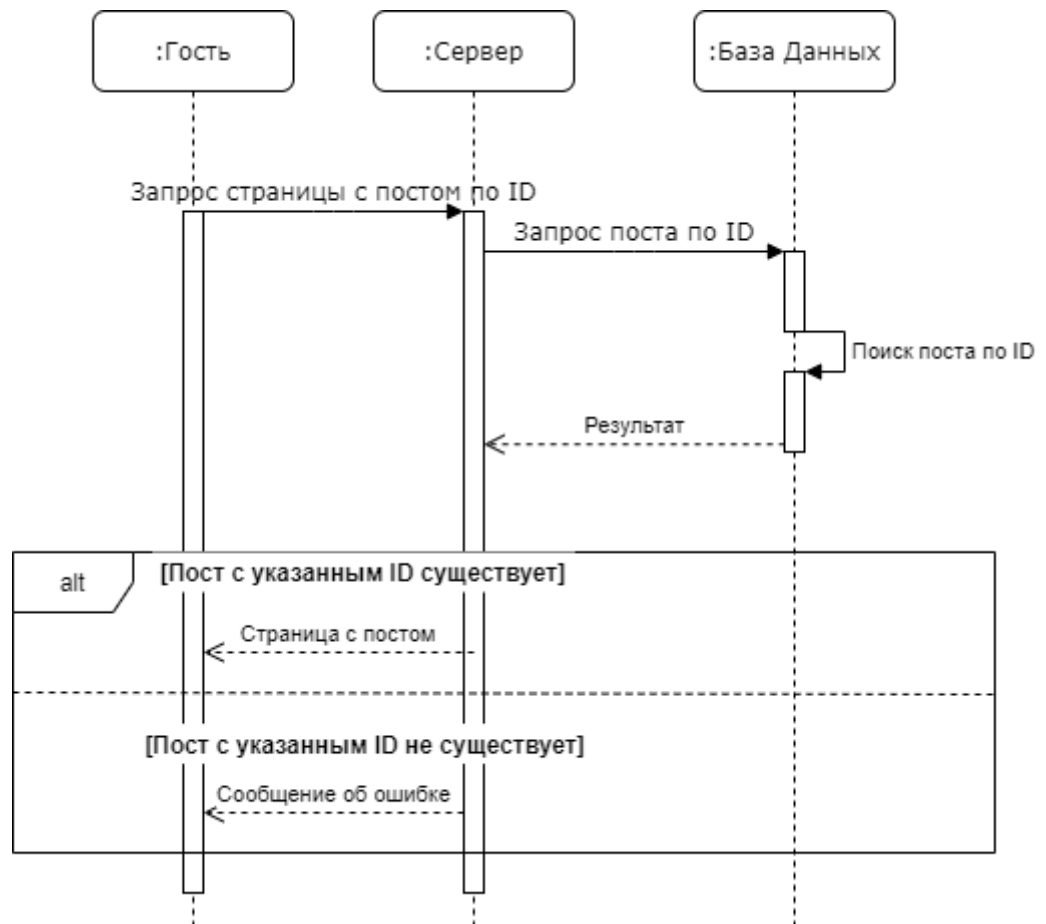
Диаграмма последовательности. Редактирование автором статьи.



**Диаграмма последовательности. Просмотр пользователем
страницы автора.**



**Диаграмма последовательности. Просмотр пользователем
страницы с содержанием статьи.**



ПРИЛОЖЕНИЕ Ж

Диаграмма активностей. Авторизация пользователя в системе.

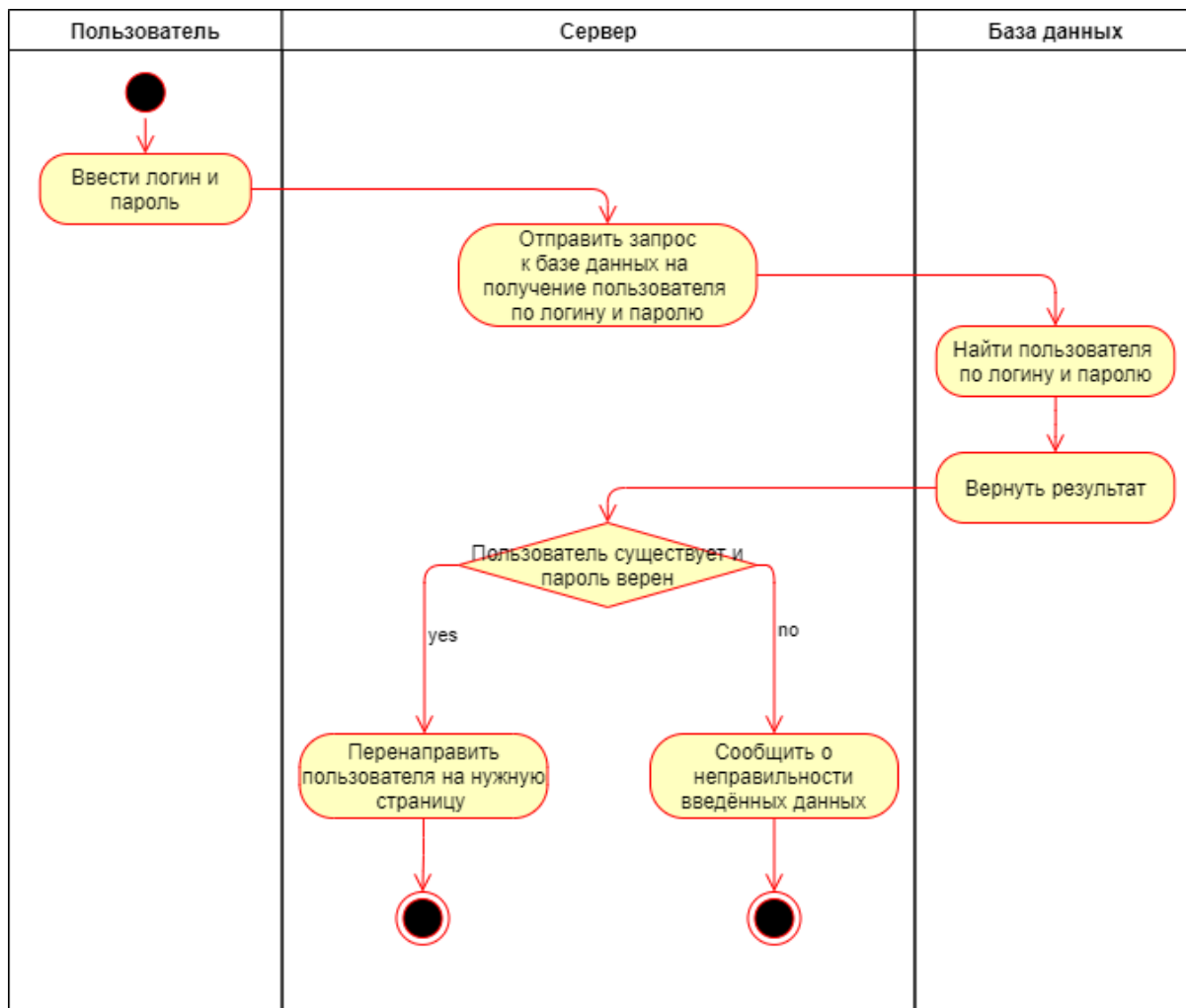


Диаграмма состояний. Взаимодействие пользователя с системой.

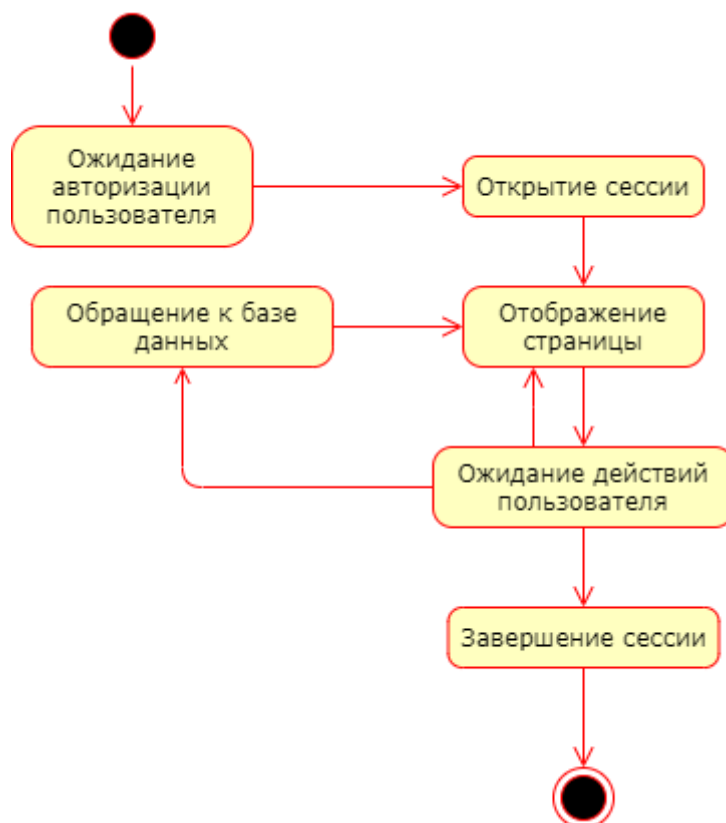
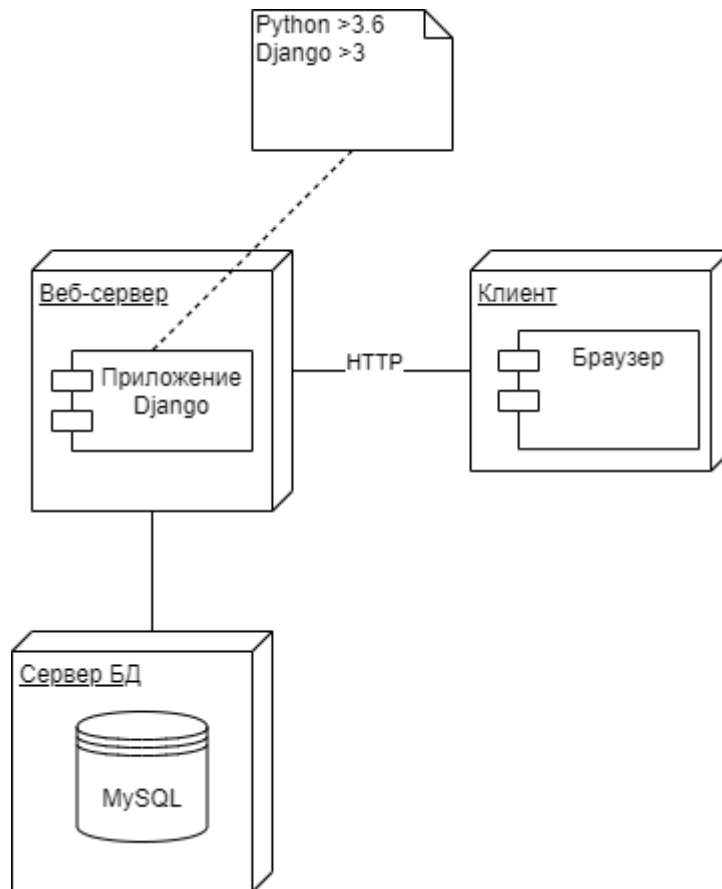

















Диаграмма развертывания



Скриншот воронок аналитики и Яндекс Метрики.

Название цели	Описание	Номер цели
1  Просмотр гостем статьи	 составная цель Переход на главную страницу: • url: содержит «nebezdari.ru» Нажатие на кнопку фильтрации постов: • событие: идентификатор цели «filter» Переход на страницу статьи: • url: начинается с «nebezdari.ru/post/»	99337300  
2  Подписка гостей на рассылку	 составная цель Переход на главную страницу : • url: совпадает «nebezdari.ru» • url: совпадает «nebezdari.ru/» Переход на страницу поста: • url: содержит «nebezdari.ru/post/» Заполнение формы подписки: • событие: идентификатор цели «subscribe_form» Нажать кнопку "Подписаться": • событие: идентификатор цели «subscribe_button»	99337558  
3  Авторизация авторов	 составная цель Переход на главную страницу: • url: совпадает «nebezdari.ru» • url: совпадает «nebezdari.ru/» Переход на страницу авторизации: • url: содержит «nebezdari.ru/login/» • url: содержит «nebezdari.ru/author/login» Ввести данные: • событие: идентификатор цели «login_form» Нажать кнопку "Авторизоваться": • url: содержит «login_button»	99338365  
4  Добавление автором новой статьи	 составная цель Переход на страницу добавления статьи: • url: содержит «nebezdari.ru/post/add» Заполнить поля: • событие: идентификатор цели «post_form» Нажать кнопку "Опубликовать": • событие: идентификатор цели «post_button»	99338800  

Сохранить

