МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра информационных систем

Техническое задание

на разработку веб-приложения

«Сервис для создания и управления персонализированными новостными лентами, с возможностью фильтрации контента по интересам «Мои Новости»»

Исполнители

А. О. Поляков

Д. Д. Рындин

В.Д. Михайлов Заказчик

В.С. Тарасов

Воронеж 2024

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc161827486)

[1 ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ 4](#_Toc161827487)

[2 Общие положения 6](#_Toc161827488)

[2.1 Название приложения 6](#_Toc161827489)

[2.2 Наименование разработчика и заказчика и их реквизиты 6](#_Toc161827490)

[2.3 Перечень документов, на основании которых создаётся приложение 6](#_Toc161827491)

[2.4 Плановые сроки начала и окончания работ по созданию АС 7](#_Toc161827492)

[2.5 Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения 7](#_Toc161827493)

[3 Цели и назначение создания автоматизированной системы 9](#_Toc161827494)

[3.1 Цели создания АС 9](#_Toc161827495)

[3.2 Назначение АС 9](#_Toc161827496)

[4 Требования к веб-приложению и программному обеспечению 10](#_Toc161827497)

[4.1 Требования к программному обеспечению веб-приложения 10](#_Toc161827498)

[4.2 Структура веб-приложения 11](#_Toc161827499)

[4.3 Группы пользователей 11](#_Toc161827500)

[4.4 Требования к квалификации персонала, обслуживающего веб-приложение 12](#_Toc161827501)

[4.5 Языковые требования веб-приложения 12](#_Toc161827502)

[5 Дизайн веб-приложения 13](#_Toc161827503)

[6 Навигация по приложению 14](#_Toc161827504)

[7 Функциональность веб-приложения 16](#_Toc161827505)

[7.1 Функциональные возможности для всех пользователей 16](#_Toc161827506)

[7.1.1 Регистрация и авторизация пользователя 16](#_Toc161827507)

[7.1.2 Просмотр новостей и журналов 16](#_Toc161827508)

[7.1.3 Система тегов. 17](#_Toc161827509)

[7.2 Дополнительные функциональные возможности авторизованных пользователей 18](#_Toc161827510)

[7.3 Функциональные возможности редактора 18](#_Toc161827511)

[7.4 Функциональные возможности администратора 18](#_Toc161827512)

[ПРИЛОЖЕНИЕ А 20](#_Toc161827513)

[ПРИЛОЖЕНИЕ Б 23](#_Toc161827514)

[ПРИЛОЖЕНИЕ В 26](#_Toc161827515)

[ПРИЛОЖЕНИЕ Г 27](#_Toc161827516)

[ПРИЛОЖЕНИЕ Д 28](#_Toc161827517)

[ПРИЛОЖЕНИЕ E 29](#_Toc161827518)

[ПРИЛОЖЕНИЕ Ж 30](#_Toc161827519)

1. ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ

**Веб-приложение** - программное изделие, разновидность прикладного [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), предназначенная для работы на смартфонах, [планшетах](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD%D1%88%D0%B5%D1%82%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и других мобильных (портативных, переносных, карманных) устройствах.

**БД** – это организованная коллекция данных, хранящихся в централизованном месте и структурированные таким образом, чтобы обеспечивать эффективное добавление, доступ, управление и обновление информации.

**Frontend** – презентационная часть информационной или программной системы, ее пользовательский интерфейс и связанные с ним компоненты.

**Backend** – логика работы сайта, внутренняя часть продукта, которая находится на сервере и скрыта от пользователя.

**Клиент, клиентская сторона** – приложение, которое предоставляет пользователю возможность взаимодействовать со всей системой.

**Сервер, серверная часть** – компьютер, обслуживающий другие устройства (то есть, клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

**GitHub** – веб-сервис для хостинга IT-проектов и их совместной разработки.

**SQLite** – реляционная БД с открытым кодом.

**Фреймворк** – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

**Python** - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.

**АС** – автоматизированная система; организационно-техническую система, обеспечивает выработку решений через автоматизацию информационных процессов в различных сферах деятельности.

**REST API (REST)** – стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений.

**Django** – высокоуровневый Python веб–фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб–сайты.

**СУБД** – набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в БД.

**JavaScript** – язык программирования, предназначенный для front-end-разработки и использующийся для взаимодействия с пользователем.

**CSS** – формальный язык описания внешнего вида веб-страницы, написанного с использованием языка разметки.

**HTML** – стандартизированный язык разметки для просмотра веб-страниц в браузере.

**Vue.js** – JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов.

1. Общие положения
   1. Название приложения

Полное наименование приложения: «Сервис для создания и управления персонализированными новостными лентами, с возможностью фильтрации контента по интересам «Мои Новости»».

Условное обозначение приложения: «Мои Новости».

* 1. Наименование разработчика и заказчика и их реквизиты

Заказчик: Старший Преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: «3» команда группы «4». Состав команды разработчика:

* Михайлов Вадим Дмитриевич, Backend developer, Руководитель Проекта;
* Поляков Артем Олегович, Frontend developer, QA-инженер;
* Рындин Денис Дмитриевич, Бизнес аналитик, Дизайнер.
  1. Перечень документов, на основании которых создаётся приложение

Приложение создаётся на основе:

* Федерального закона «Об информации, информационных технологиях о защите информации» от 27.07.2006 г. № 149-ФЗ;
* федерального закона "О персональных данных" от 27.07.2006 N 152-Ф3.
  1. Плановые сроки начала и окончания работ по созданию АС

Состав и содержание работ по созданию сайта включают в себя следующие этапы:

* Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы 12.02.24 – 27.02.24;
* Анализ предметной области, анализ системы и построение структуры требований, ведущих к решению поставленных задач и целей до 13.03.24;
* Разработка курсового проекта, содержащего аналитическую информацию о сервисе 13.03.24 – 31.05.24;
* Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели базы данных до 25.03.24;
* Построение рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы до 17.04.24;
* Проведение тестирования и доработка программного обеспечения (сервиса) до 31.05.24.
  1. Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

* 1 аттестация (конец марта 2024) - создан репозиторий проекта на GitHub, создан проект в Miro с описанием системы, предоставлено техническое задание, предоставлены промежуточные результаты;
* 2 аттестация (конец апреля 2024) - написана основополагающая часть кода веб-приложения, реализовано взаимодействие сервера с базой данных, проведена отладка кода, доработка кода, проведено тестирование по работе сервиса;
* 3 аттестация (конец мая 2024) - разработан курсовой проект, выполнены завершающие работы по доработке веб-приложения, предоставлена готовая система.

Результаты работы должны быть представлены в электронном виде в формате PDF и размещены на GitHub.

1. Цели и назначение создания автоматизированной системы
   1. Цели создания АС

Целями создания приложения являются:

* Реализация сервиса, позволяющего просматривать новости с возможностью выбора тегов для пользователей данного сервиса (разные категории пользователей);
* Реализация получения актуальной информации из журналов от редакторов;
* Реализация возможности просмотра таргетированных новостей для пользователей и гостей.
  1. Назначение АС

Приложение позволяет решать следующие задачи:

* Авторизация, регистрация аккаунта, получение и редактирование данных об аккаунте;
* Просмотр новостей или журналов по выбранному тегу;
* Настройка тегов в качестве авторизованного пользователя;
* Просмотр таргетированных новостей или журналов;
* Создание, редактирование, удаление новостей и журналов в качестве редактора;
* Назначение редакторов в качестве администратора;
* Редактирование и удаление аккаунтов в качестве администратора.

1. Требования к веб-приложению и программному обеспечению

Веб-приложение должно следовать архитектуре, соответствующую смешанной модели клиент - серверного взаимодействия (описано в ПРИЛОЖЕНИЕ Ж) на основе REST API. Приложение должно корректно и одинаково работать на различных браузерах.

* 1. Требования к программному обеспечению веб-приложения

Для реализации серверной части приложения будут использоваться следующие средства:

* Django. Данный фреймворк является одним из основных используемых фреймворков для разработки веб-приложений. Также данный фреймворк отличается выделенной структурой для приложения. Также Django имеет возможности для защиты данных, а также подходит для разработки большинства типов веб-приложений;
* Python. Основными преимуществами Python является наличие большого числа библиотек и фреймворков. Также язык Python ввиду нескольких применяемых парадигм программирования подходит для написания различных видов приложений, в том числе и веб-приложений. Также одним из важных преимуществ Python является работа почти на любой операционной системе или платформе;
* СУБД SQLite. Данная СУБД является свободно распространяется. Также она может располагаться локально на любом устройстве. Для работы с SQLite не требуется широкий набор библиотек, потому она является одной из самых простых в использовании. Также данная СУБД отличается надежностью и эффективностью, что позволяет использовать её в веб-приложениях.

Для реализации клиентской части приложения будут использоваться следующие средства:

* Язык программирования JavaScript. Данный язык программирования поддерживается в абсолютном большинстве браузеров, что позволяет работать с ним вне зависимости от конкретной операционной системы или платформы. Является основным языком сценариев для браузеров.
* Vue.js. JavaScript-фреймворк для создания пользовательских интерфейсов. Используется для обработки веб-страниц в реактивном стиле. Обладает кроссплатформенностью, то есть, поддерживается в тех случаях, где будет поддерживаться JavaScript. Позволяет писать проекты с учетом модульности приложений.
  1. Структура веб-приложения

Данный сервис должен разрабатываться на основе модели клиент-серверного взаимодействия на основе REST API.

Система должна представлять собой серверную и клиентскую части (backend и frontend). Серверная часть выполняет обработку запросов, работу с базой данных и логику, клиентская часть выполняет функции отображения информации на веб-странице.

* 1. Группы пользователей

Сервис предусматривает наличие следующих ролей:

* Неавторизованный пользователь, гость – лицо без учетной записи, с возможностью авторизации и регистрации, а также с возможностью просмотра статей и журналов;
* Авторизованный пользователь – авторизованный в системе человек, имеющий возможности редактирования информации об аккаунте, выбора тегов;
* Редактор – пользователь с возможностью создания, редактирования, удаления новостей и журналов;
* Администратор – пользователь с правами на назначение редакторов, удаление аккаунтов, создание, удаление и редактирование новостей и журналов.
  1. Требования к квалификации персонала, обслуживающего веб-приложение

Обслуживающий данный сервис – администратор. Основными работниками, создающими контент, являются редакторы. Редакторы – квалифицированные работники с возможностью работы с публикациями (новостями и журналами). Квалификация определяется наличием языковых навыков, знаний общепринятых стандартов написания текста, умения работать с текстом. Администратор назначает редакторов в сервисе.

* 1. Языковые требования веб-приложения

В текущее время (на этапе создания требований и проектирования) не планируется поддержка других языков, помимо русского.

1. Дизайн веб-приложения

Присутствуют определенные базовые правила, которых следует придерживаться:

* Веб-приложение должно быть выполнено в едином стиле.
* Обязательно наличие не более трех шрифтов и умеренная цветовая палитра.
* Должна быть реализована адаптивная вёрстка под разные размеры окон.
* В левой части страницы должна присутствовать навигационная панель.
* На веб-странице объемный текст может быть только на страницах с журналами и новостями, а также на специальных страницах с информацией.

1. Навигация по приложению

Данный раздел представлен в виде навигационного меню в левой части веб-страницы с определенными подразделами. При нажатии на кнопки меню происходит переход на соответствующую страницу приложения.

Для неавторизованного пользователя меню содержит следующие подразделы:

* Главная страница с публикациями;
* Страница конкретной публикации;
* Страница регистрации;
* Страница авторизации.

Для авторизованного пользователя меню содержит следующие подразделы:

* Главная страница с публикациями;
* Страница выбора тегов;
* Страница с подпиской;
* Страница конкретной публикации;
* Страница с журналами;
* Страница с конкретным журналом;
* Личный кабинет.

Для редактора меню содержит следующие подразделы:

* Главная страница с публикациями;
* Страница с подпиской;
* Страница конкретной публикации;
* Страница с журналами;
* Страница с конкретным журналом;
* Страница с созданием новости/журнала;
* Просмотр статистики публикаций;
* Личный кабинет.

Для администратора меню содержит следующие подразделы:

* Страница с назначением редакторов;
* Страница с подпиской;
* Страница конкретной публикации;
* Страница с журналами;
* Страница с конкретным журналом;
* Страница со списком пользователей;
* Личный кабинет.

1. Функциональность веб-приложения

Сервис предоставляет некоторый ряд функций по запросам просмотра новостей и журналов у пользователей. Функциональность сервиса отражена в функциональной схеме и диаграмме прецедентов (ПРИЛОЖЕНИЕ А и ПРИЛОЖЕНИЕ Б соответственно).

* 1. Функциональные возможности для всех пользователей
     1. Регистрация и авторизация пользователя

При первичном использовании веб-приложения у гостя есть возможность зарегистрироваться на сайте (регистрация, авторизация и дальнейшие действия описаны в ПРИЛОЖЕНИЕ В).

* При первом входе отображается главная страница сайта. Неавторизованный пользователь может просмотреть только 10 новостей, при достижении этого количества страница уведомляет гостя о лимите.
* При нажатии на кнопку регистрации происходит переход на страницу регистрации. На ней проверяются введенные данные, при успешной регистрации создаётся новый пользователь и осуществляется переход на главную страницу сайта.
* При нажатии на кнопку авторизации происходит переход на страницу авторизации. Аналогично странице с регистрацией, проверяются введенные данные, при успешной регистрации происходит вход в аккаунт и переход на главную страницу сайта.
  + 1. Просмотр новостей и журналов

При нажатии на конкретную новость происходит переход на конкретную страницу с новостью (взаимодействие системы внутри себя и с пользователем в ПРИЛОЖЕНИЕ Г, структура БД с публикацией в ПРИЛОЖЕНИЕ Д).

* При достижении лимита неавторизованным пользователем переход на новость становится невозможным до регистрации или авторизации.
* При переходе к странице новости происходит сбор статистики. На отображенной странице находятся название, текст и прикрепленный файл новости.
* Просмотр журналов возможен только при авторизации. При переходе на страницу журнала, аналогично новости, отображаются название, текст и прикрепленный файл журнала.
  + 1. Система тегов.

Выбор тега происходит на соответствующей отдельной странице сайта.

* При нажатии на конкретный тег он выделяется, на странице с новостями или журналами отображаются только те новости или журналы, у которых есть этот тег. При повторном нажатии на тег он перестает быть выделенным, и никак не влияет на подбор новостей или журналов.

Для неавторизованного пользователя теги с просмотренных новостей сохраняются, выше отображаются новости с просматриваемыми тегами (с учетом лимита).

Для авторизованного пользователя просмотренные теги также сохраняются, но таргетированная система включает больше возможностей, так как отсутствует лимит на количество просмотренных новостей, а также начинают отображаться журналы, аналогично отображаясь выше с учетом тегов.

Также для авторизованных пользователей есть возможность выбора локального тега, то есть, тега для определенного населенного пункта.

* 1. Дополнительные функциональные возможности авторизованных пользователей

Авторизованные пользователи могут осуществлять следующее взаимодействие с публикациями и тегами:

* Сохранение в избранное новости или журналы;
* Поиск новостей или журналов по названию или тегу;
* Оформление подписки на издателя.
  1. Функциональные возможности редактора

При нажатии на взаимодействии с публикацией (новостью или журналом) редактор переходит на страницу с редактированием соответствующей публикации.

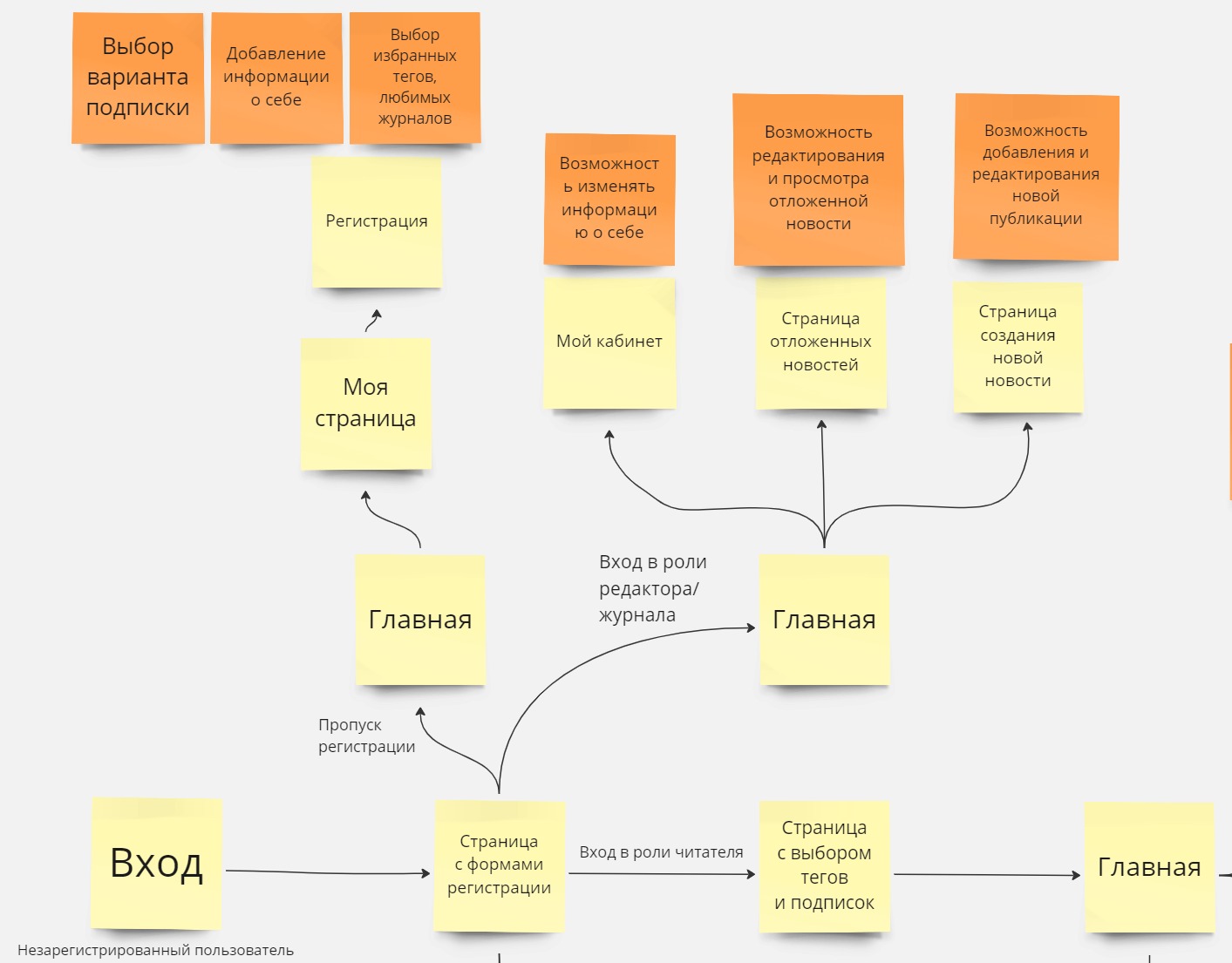
* При нажатии на кнопку создании публикации редактор переходит на соответствующую страницу редактирования. Поля публикации не заполнены. Также редактор может выбрать время публикации.
* При нажатии на кнопку редактирования публикации редактор, аналогично созданию, переходит на страницу редактирования. В этом случае поля публикации заполнены, статистика публикации сохраняется. Также есть возможность выбора времени публикации.
* При нажатии на кнопку удаления публикации она удаляется вместе со статистикой.
* При нажатии на кнопку просмотра статистики журналов в личном кабинете редактор переходит на соответствующую страницу со статистикой своих публикаций.
  1. Функциональные возможности администратора

Функциональные возможности аналогичным таковым у редактора. Помимо этого, администратор обладает возможностями:

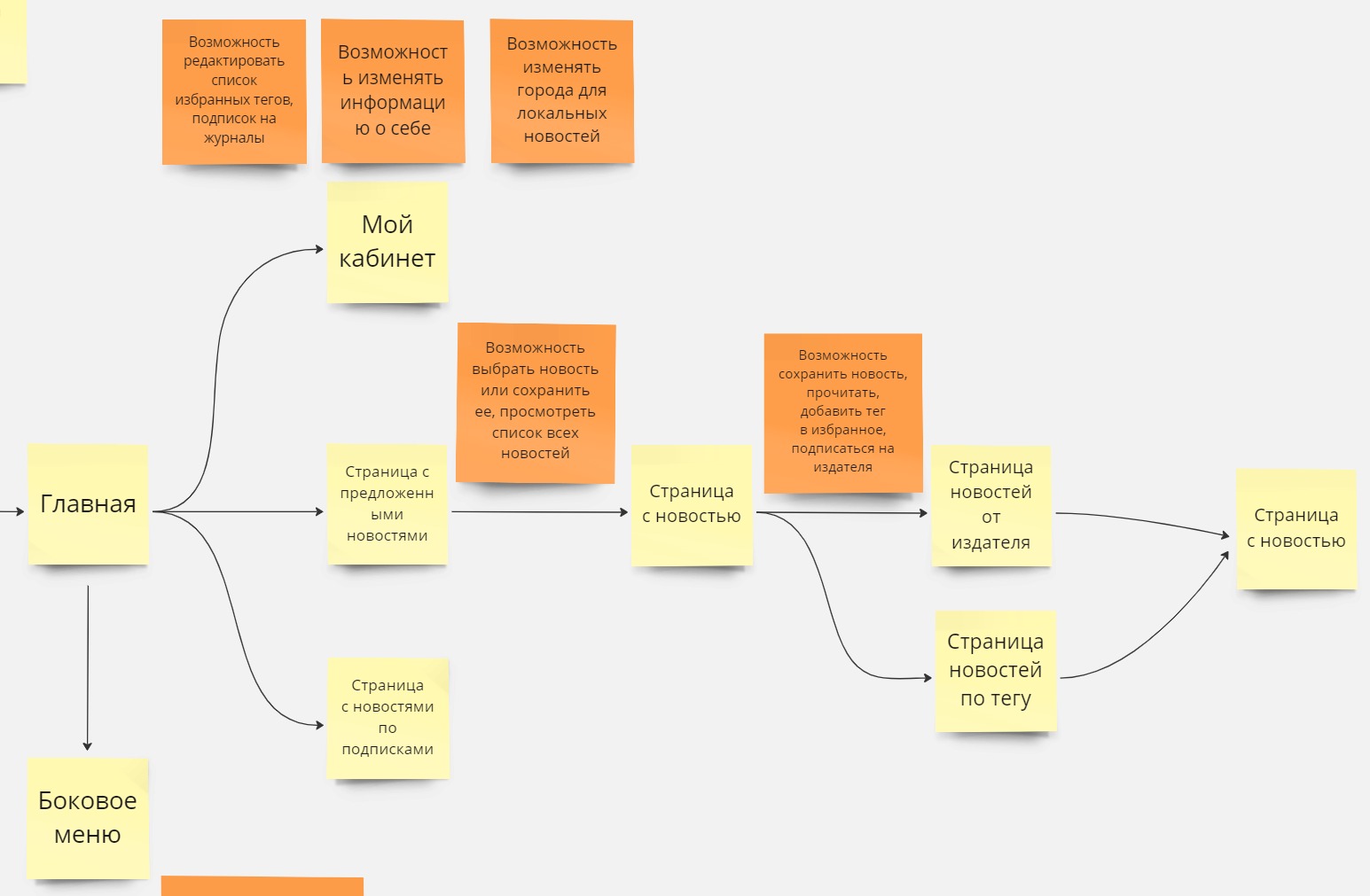
* Назначать и убирать роли редакторов (процесс выдачи роли редактора описан в ПРИЛОЖЕНИЕ Е);
* Удаление любого пользователя из соответствующего списка.

Также администратор может редактировать и удалять контент аналогично редактору, имея доступ к любому контенту.

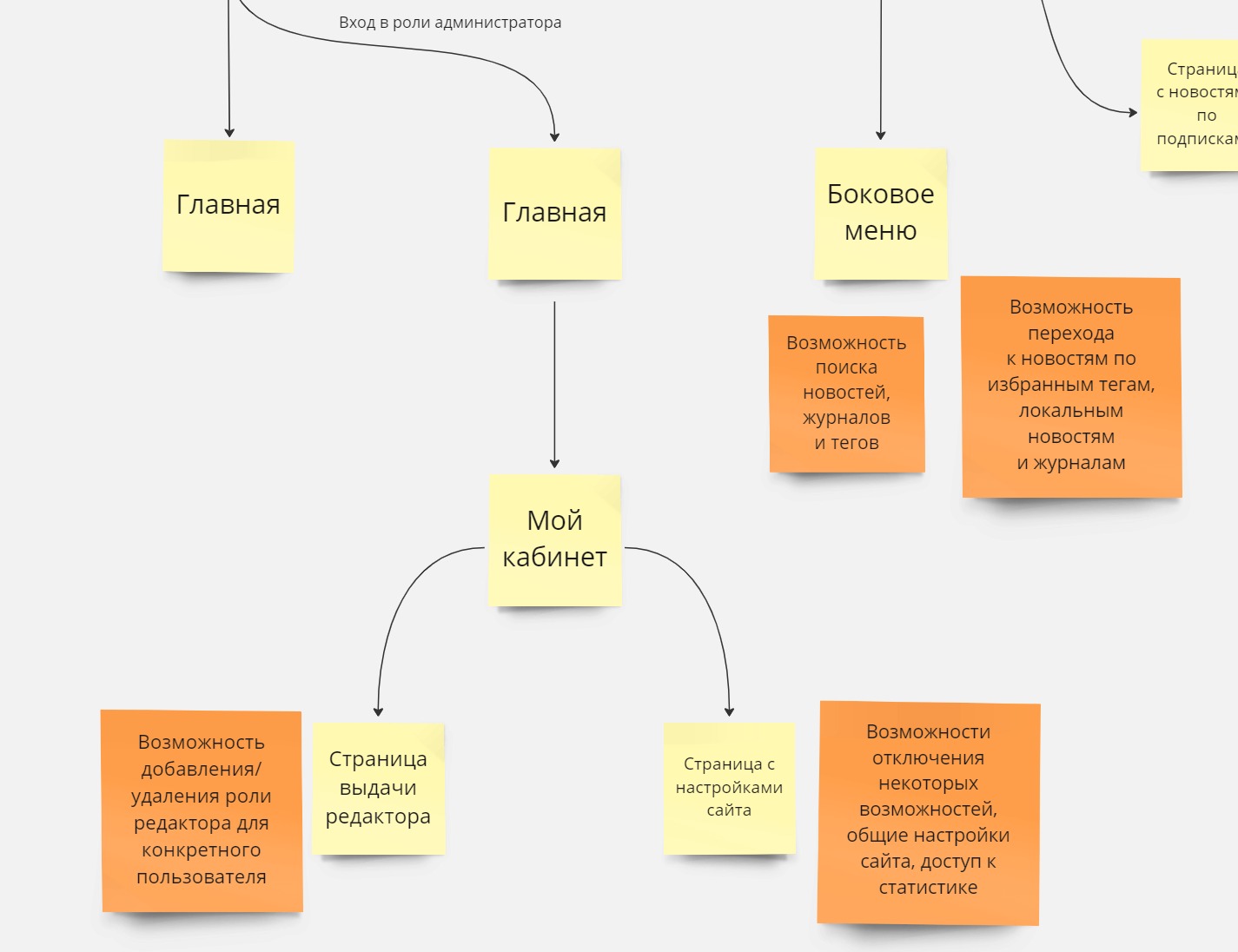
ПРИЛОЖЕНИЕ А



1. Функциональная схема приложения, часть 1

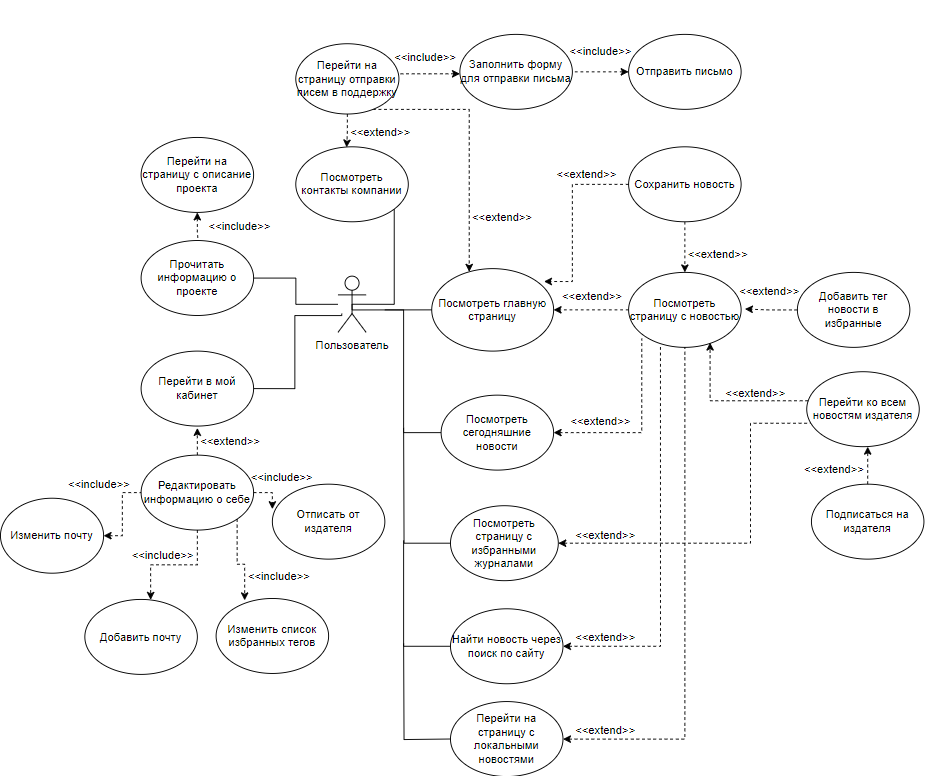


1. Функциональная схема приложения, часть 2

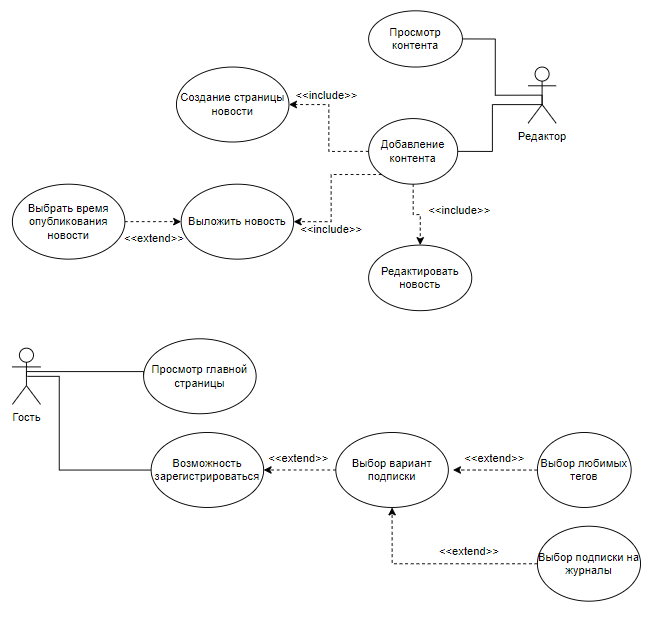


1. Функциональная схема приложения, часть 3

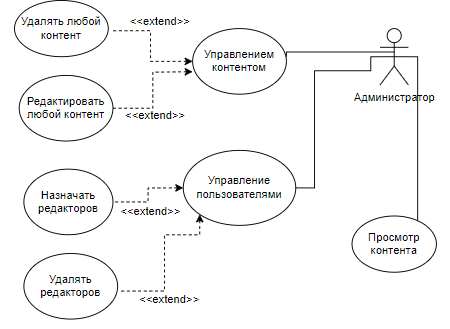
ПРИЛОЖЕНИЕ Б



1. Диаграмма прецедентов - пользователь

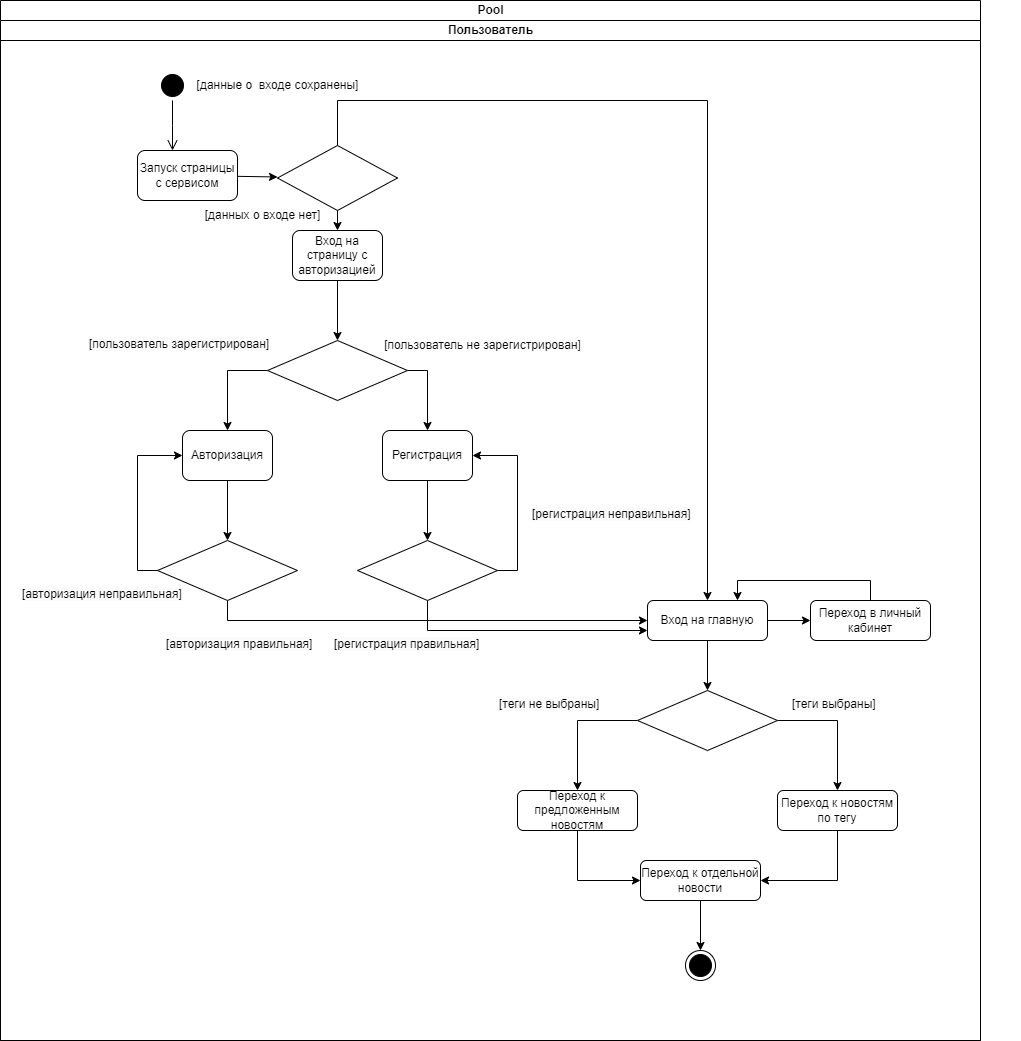


1. Диаграмма прецедентов – редактор, гость



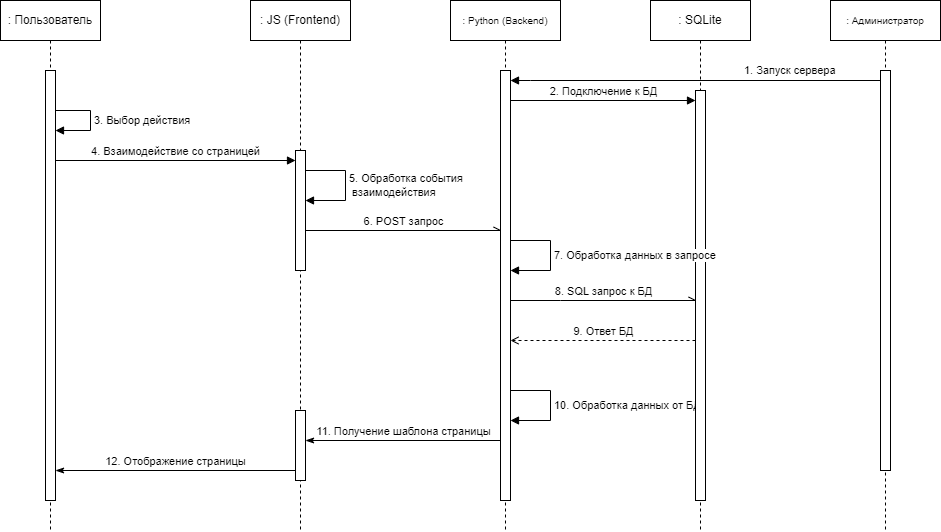
1. Диаграмма прецедентов - администратор

ПРИЛОЖЕНИЕ В



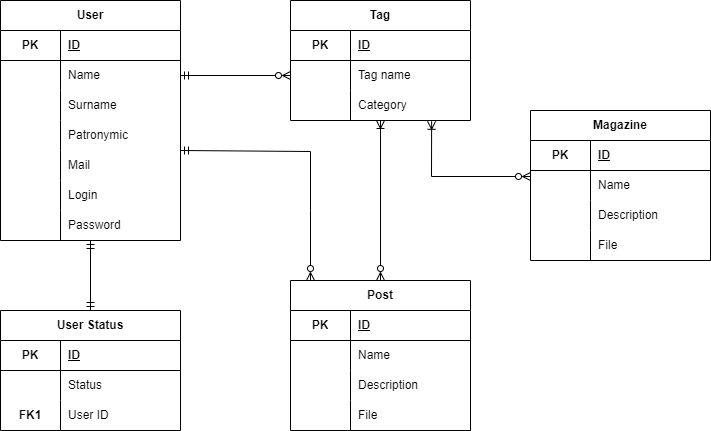
1. Диаграмма активности для взаимодействия пользователя с сервисом (регистрация, авторизация, просмотр)

ПРИЛОЖЕНИЕ Г



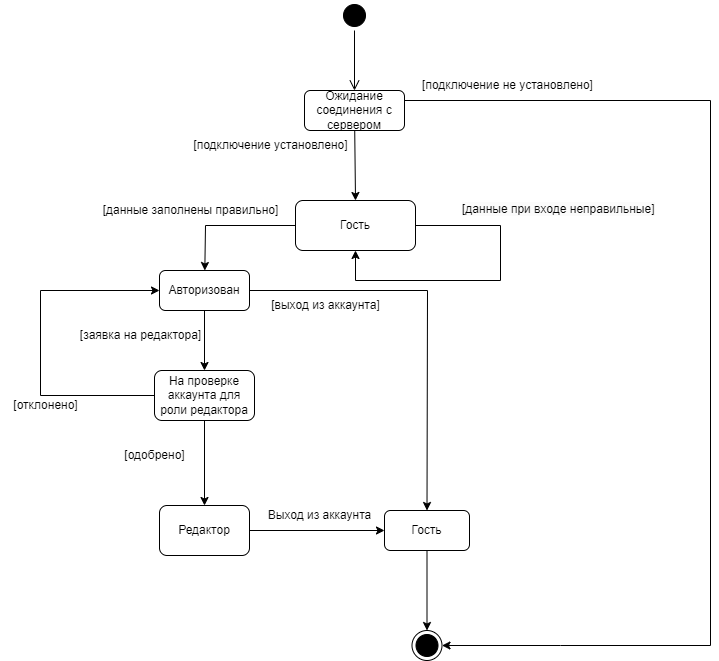
1. Диаграмма последовательности для взаимодействия пользователя и системы

ПРИЛОЖЕНИЕ Д



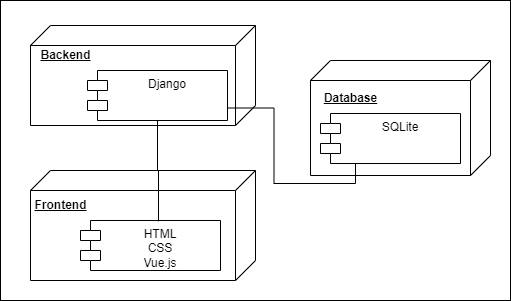
1. ER-диаграмма, описывающая структуру БД

ПРИЛОЖЕНИЕ E



1. Диаграмма состояний, описывающая процесс выдачи роли редактора

ПРИЛОЖЕНИЕ Ж



1. Диаграмма развертывания, описывающая структуру взаимодействия Frontend части, Backend части и БД