

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Разработка и реализация информационной системы портала

«Милая страничка»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Ульяновой Олеси Павловны

Научный руководитель
доцент, к.ф.-м.н., доцент

О.С. Кузнецова

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

Введение. В современном мире информационные технологии играют значимую роль в сохранении и популяризации культурного наследия. Благодаря развитию веб-технологий, создаются платформы, позволяющие не только архивировать произведения искусства, но и предоставлять пользователям доступ к ним в удобном цифровом формате. Особенно актуальной эта тенденция становится в сфере литературы, где информационные системы помогают сделать произведения авторов доступными широкой аудитории.

Портал «Милая страничка» представляет собой информационную систему, направленную на сохранение стихотворного наследия моего деда — поэта, а также на популяризацию культурной жизни Сосновоборского района. Сайт обеспечит доступ к биографической информации, стихам автора, новостям о культурных событиях района. Информационная система будет включать функциональные элементы, способствующие удобному взаимодействию пользователей с контентом.

Актуальность данной работы обусловлена необходимостью сохранения и популяризации литературного наследия с использованием современных информационных технологий.

Целью данной работы является создание информационной системы, которая станет платформой для представления литературного наследия автора и культурной жизни района. Для достижения поставленной цели необходимо решить следующие задачи:

- Изучение теоретических аспектов разработки информационных систем;
- Анализ предметной области;
- Проектирование структуры информационной системы;
- Проектирование и реализация базы данных информационной системы (ИС);
- Разработка и реализация серверной и клиентской части ИС.

Первый раздел содержит информацию об информационных системах, их предназначении и о способах построения архитектур ИС, а также включает информацию о предметной области.

ИС представляет собой сложный комплекс программных, аппаратных, организационных и методических средств, предназначенных для автоматизированной обработки данных. Она обеспечивает сбор, хранение, анализ, передачу

и представление информации, необходимой для поддержки процессов принятия решений, управления ресурсами и оптимизации рабочих процессов.

Архитектура ИС определяет ее внутреннее устройство, распределение функциональных компонентов и способы взаимодействия между ними. Архитектуру ИС можно представить в виде следующих компонентов:

1. Слой представления (Frontend).
2. Слой бизнес-логики (Backend).
3. Слой доступа к данным.

В данной работе применяется трехуровневая архитектура клиент-сервер, что позволяет создать более гибкую и масштабируемую систему. Основные компоненты архитектуры включают: сервер, систему управления базами данных PostgreSQL, сервер приложений Waitress и клиентскую часть, представляемую веб-браузером.

Предметной областью данной информационной системы является творчество поэта Ивана Афанасьевича Ульянова, представленное на портале «Милая сторона». Целью создания сайта является обеспечение удобного доступа пользователей к произведениям автора, его биографии, фотографиям и актуальным событиям, связанным с литературной деятельностью.

Основными компонентами предметной области «Милая сторона» являются: контентная составляющая (литературные материалы), пользователи системы, контентные процессы, участники системы, технологическая база.

На сайте реализованы следующие разделы: «Главная», «Биография», «Каталог», «Фотографии», «События».

Второй раздел посвящён проектированию информационной системы с помощью языка UML (Unified Modeling Language). Он содержит теоретическую часть, которая предоставляет основную информацию об UML, и практическую, в которой осуществляется проектирование ИС посредством построения трёх диаграмм: диаграммы вариантов использования, диаграммы последовательности, диаграммы активности.

В рамках разработки информационной системы «Милая сторона» использование UML позволяет:

- Определить ключевые компоненты (пользователи, материалы поэта, подписка);

- С моделировать взаимодействие пользователей (просмотр стихов, покупка подписки);
- Проанализировать альтернативные сценарии (ошибки оплаты, регистрация);
- Определить сущности базы данных и их связи.

Диаграмма вариантов использования демонстрирует, какие функции выполняет система с точки зрения пользователя. В системе «Милая страничка» основные варианты использования включают:

- Ознакомление с биографией поэта;
- Чтение стихотворений;
- Просмотр фотографий;
- Просмотр событий и новостей;
- Оформление подписки.

Диаграмма последовательности (Sequence Diagram) в UML предназначена для визуального представления временного порядка взаимодействия между объектами системы. Для системы «Милая страничка», представляющей сайт поэта Ульянова И.А., можно выделить следующий сценарий: **«Покупка подписки»**.

Основной поток событий:

1. Пользователь заходит на сайт.
2. Пользователь открывает страницу каталога.
3. Выбирает подписку.
4. На странице подписки пользователь вводит свой адрес электронной почты.
5. Пользователь оплачивает подписку.
6. Система активирует подписку.

В работе представлена диаграмма последовательности для выбранной информационной системы. Результат представлен в соответствии с рисунком 1.

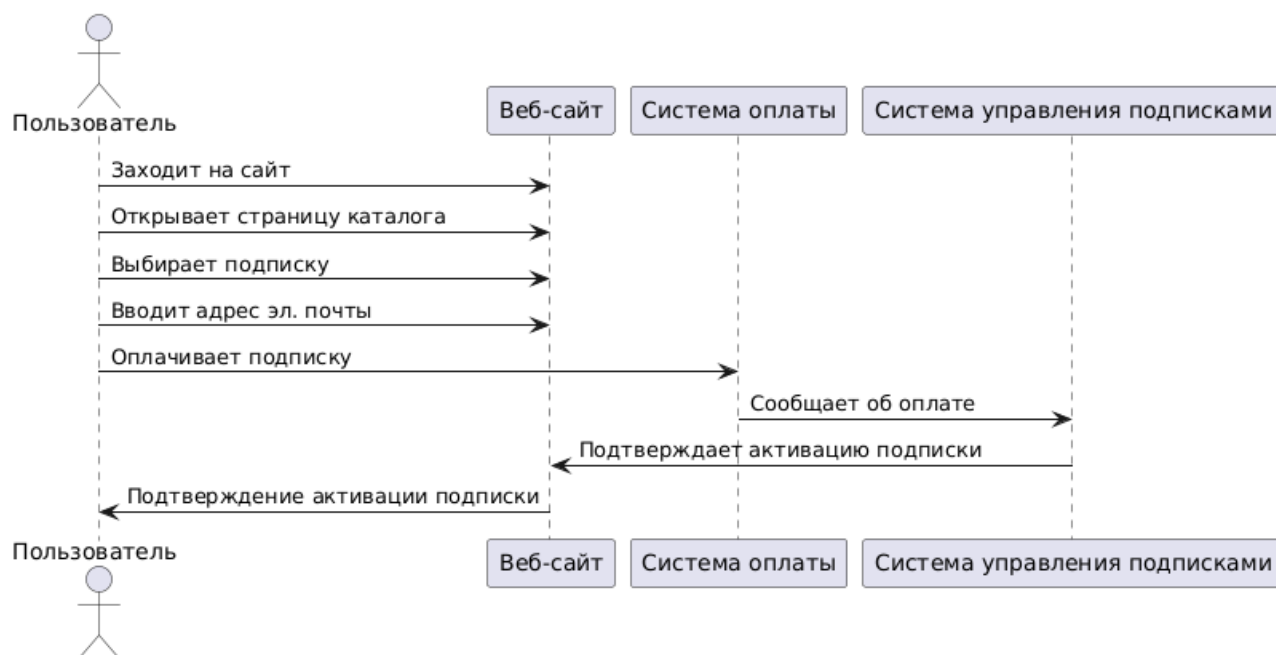


Рисунок 1 — Диаграмма последовательности для ИС «Милая сторонка»

Диаграмма активности (Activity Diagram) представляет собой графическое описание последовательности выполнения действий в системе.

Рассмотрим процесс **«Просмотр стихотворений по подписке»**:

1. Пользователь заходит на сайт.
2. Выбирает раздел «Каталог».
3. Вводит свой адрес электронной почты.
4. Система проверяет наличие подписки:
 - Если подписка активна - пользователь получает доступ.
 - Если подписки нет - система предлагает оформить подписку.
5. Пользователь читает стихи.
6. Конец процесса.

В работе представлена диаграмма активности для выбранной информационной системы. Результат представлен в соответствии с рисунком 2.

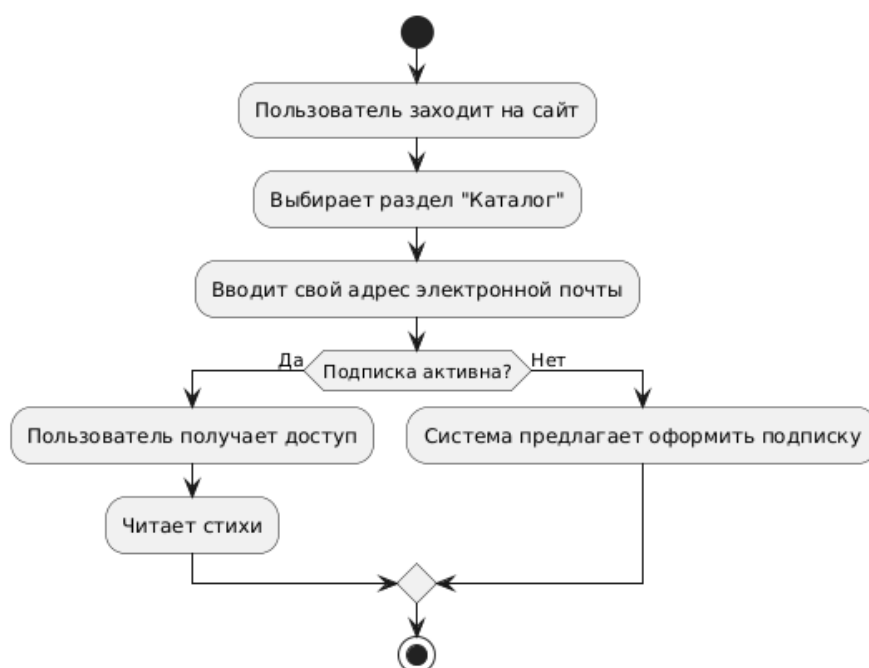


Рисунок 2 — Диаграмма активности для ИС «Милая сторонка»

Третий раздел посвящён проектированию и созданию базы данных для выбранной информационной системы.

В рамках разработки информационной системы «Милая сторонка», предназначенной для публикации и распространения творческого наследия поэта Ульянова И.А., в качестве основной СУБД (система управления базами данных) была выбрана PostgreSQL, потому что это мощная, свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), которая сочетает в себе возможности традиционных реляционных баз данных с расширенными возможностями работы со сложными типами данных. База данных PostgreSQL обеспечивает:

- Хранение текстов поэта (биография, стихи, публикации);
- Управление подписками и оплатами;
- Каталогизация медиафайлов (фотографии и метаданные);
- Управление событиями (мероприятия, встречи, литературные вечера).

База данных информационной системы «Милая сторонка» состоит из нескольких основных сущностей, каждая из которых отвечает за хранение определённых типов данных:

1. **events** – содержит новости и мероприятия, отображаемые в разделе «События».

2. **photos** – содержит изображения, которые могут быть прикреплены к событиям или отображаться отдельно.
3. **catalog_materials** – общая таблица для книг и стихов;
4. **books** – содержит книги, опубликованные в каталоге;
5. **poems** – хранит стихи, размещённые в каталоге;
6. **clients** – хранит информацию о подписчиках, их датах подписки и сроках подписки.

Связи между таблицами обеспечивают логическую целостность данных и позволяют эффективно извлекать нужную информацию.

ER - диаграмма (entity-relationship diagram) для разрабатываемой информационной системы создана с помощью сервиса draw.io и представлена в соответствии с рисунком 3.

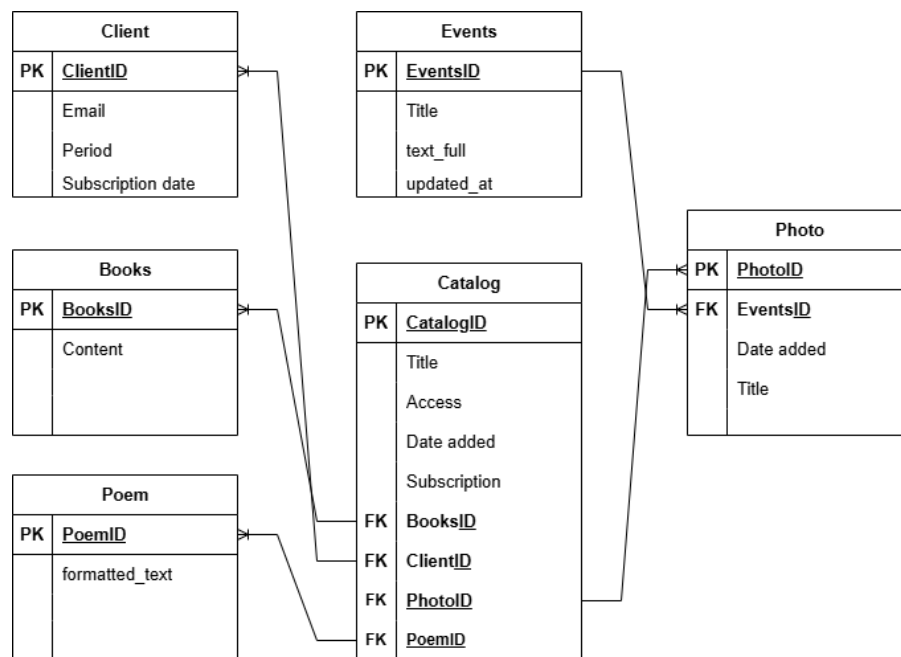


Рисунок 3 — ER-диаграмма базы данных ИС «Милая страничка»

Для взаимодействия с базой данных информационной системы «Милая страничка» используется современная библиотека SQLAlchemy 2.0.30, которая реализует технологию ORM (Object-Relational Mapping). Эта технология позволяет связать объекты Python с записями в реляционной базе данных. ORM-система абстрагирует взаимодействие с базой данных, заменяя необходимость в написании SQL-запросов, что делает разработку более удобной и понятной, а код — более компактным и легко поддерживаемым.

Для взаимодействия с базой данных мы создаём модель данных, используя классы Python. Каждый класс представляет собой таблицу в базе данных, а каждый экземпляр класса — запись в соответствующей таблице. В SQLAlchemy таблицы создаются с использованием Declarative Base, что позволяет описывать структуру базы данных в виде Python-классов.

В примере ниже представлена структура базы данных для информационной системы, которая включает таблицы `events`, `photos`, `catalog_materials`, `books`, `poems` и `clients`.

Таблицы в базе данных создаются в момент первого запуска приложения (запуск происходит через файл `__init__.py`), пример представлен в соответствии с листингом 1.

```
with app.app_context():
    db.create_all()
```

Листинг 1: Создание таблиц базы данных

Для взаимодействия с базой данных используются классы-сервисы (`CatalogService`, `ClientsService`, `EventsService`, `PhotosService`), которые работают с конкретными таблицами базы данных. Взаимодействие непосредственно с конкретными таблицами реализовано через DAO-интерфейсы (Data Access Object), пример представлен в соответствии с листингом 2.

```
from dao import ClientsDAO
clients_dao = ClientsDAO()
class ClientsService:
    def get_client_by_email(self, email):
        return clients_dao.get_client_by_email(email)

    def create_client(self, client):
        clients_dao.create_client(client)
```

Листинг 2: Взаимодействие с таблицами через DAO и сервисы

В четвёртом разделе предоставляется информация о серверной части информационной системы, для неё был использован фреймворк Flask 3.0.3, который представляет собой легковесный микрофреймворк для разработки

веб-приложений на языке Python. Этот фреймворк был выбран благодаря своей легкости, гибкости и поддержке расширений, что позволяет создавать веб-приложения с минимальными затратами на настройку и конфигурацию. Flask предоставляет множество полезных инструментов и библиотек, включая Werkzeug для работы с запросами и Jinja2 для шаблонизации HTML-страниц. Он также предоставляет возможность создания RESTful API, что является ключевым элементом нашей системы.

Flask поддерживает архитектуру RESTful, что означает использование HTTP-методов (GET, POST, PUT, DELETE) для обработки различных типов запросов. Он также поддерживает работу с токенами через `flask_jwt_extended`, что используется для обеспечения безопасности и аутентификации пользователей через JWT (JSON Web Tokens).

Структура приложения разбита на несколько ключевых контроллеров, которые управляют различными аспектами функциональности. Контроллеры обеспечивают маршрутизацию запросов и передачу данных между клиентом и сервером.

Файл `catalog_controller.py` содержит логику, связанную с каталогом материалов и подписками пользователей. В нем реализованы сервисы для работы с базой данных и предоставления информации о материалах, а также управление подписками клиентов.

- **Модели:**

- `Client` — модель, которая представляет клиентов, подписавшихся на материалы.
- `CatalogMaterial` (и её наследники `Book`, `Poem`) — модели, представляющие материалы в каталоге, такие как книги и стихи.

- **Сервисы:**

- `CatalogService` — инкапсулирует операции CRUD для работы с материалами каталога.
- `ClientsService` — обрабатывает запросы, связанные с подписками клиентов (создание, обновление, проверка подписок).

- **JWT:**

- Для управления состоянием подписки используется библиотека `flask_jwt_extended`, которая обеспечивает безопасное создание и валидацию токенов.

Для передачи данных между сервером и клиентом используется REST API. REST (Representational State Transfer) — это архитектурный стиль взаимодействия, который использует стандартные HTTP-методы (`GET`, `POST`, `PUT`, `DELETE`) для выполнения операций с данными.

Графический интерфейс информационной системы был разработан с использованием веб-технологий HTML, CSS, JavaScript (jQuery), что позволило создать интуитивно понятный и удобный пользовательский интерфейс. Для рендеринга динамического контента страниц применен встроенный в Flask шаблонизатор Jinja2, который позволяет отделить логику отображения данных от бизнес-логики.

Графический интерфейс веб-приложения был организован следующим образом:

1. **Каталог с шаблонами:** Все страницы веб-приложения размещены в каталоге `webapp/templates`. Это место хранения всех HTML-шаблонов, которые обрабатываются сервером Flask с использованием Jinja2.
2. **Includes:** В папке `webapp/templates/includes` хранятся общие элементы интерфейса, которые используются на нескольких страницах:
 - **header** — верхняя часть страницы, где подключаются все стили и JavaScript-скрипты. Она обеспечивает унифицированный вид сайта, включая элементы навигации, логотип и другие важные компоненты интерфейса.
 - **footer** — нижняя часть сайта, которая содержит общую информацию, ссылки на страницы, контактные данные и прочее.
 - **up-arrow** — элемент страницы, представляющий собой стрелку, позволяющую пользователю быстро вернуться в верхнюю часть страницы при прокрутке вниз.
3. **Страницы:**
 - **biography** — страница, на которой отображается биография автора сайта.

- **catalog** — каталог с материалами, доступными только по подписке. На этой странице пользователи могут просматривать доступные материалы, если они подписались на соответствующие услуги.
- **events** — страница новостей сайта, которая отображает актуальные события, связанные с деятельностью сайта.
- **photos** — страница с фотографиями, где пользователи могут просматривать изображения, прикрепленные к событиям и материалам каталога.
- **index** — главная страница сайта, которая содержит общую информацию о проекте и ссылки на другие разделы.

Главная страница сайта представлена в соответствии с рисунком 4. На ней предусмотрена навигация, располагающаяся в верхней части страницы. Она включает кнопки для перехода в различные разделы: «Главная», «Биография», «Каталог», «Фотографии» и «События». Это обеспечит пользователям удобный доступ к ключевым частям сайта, позволяя быстро ориентироваться в представленной информации.

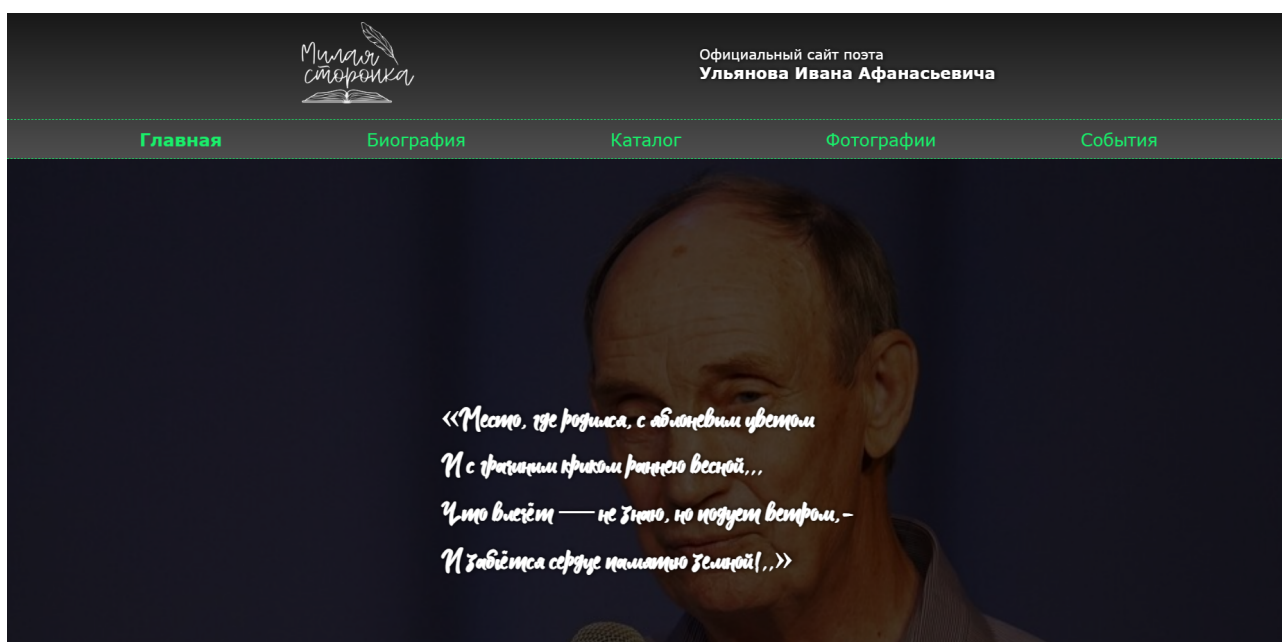


Рисунок 4 — Главная страница

Заключение. В рамках бакалаврской работы была разработана информационная система «Милая сторонка» для сохранения и популяризации стихотворного наследия моего деда — поэта, а также для развития культурной

жизни Сосновоборского района. Платформа предоставляет доступ к стихам, биографии автора и новостям, а также позволяет пользователям обсуждать произведения и следить за событиями.

Система включает серверную часть на Flask, базу данных на SQLAlchemy и графический интерфейс на HTML, CSS, jQuery с адаптивным дизайном и параллакс-скроллингом. Передача данных осуществляется через REST API.

Цели и задачи, поставленные в рамках работы, были успешно достигнуты. В результате проделанной работы была создана полноценная информационная система, которая служит важным инструментом для активного вовлечения пользователей в обсуждение.