Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра ФиПМ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине «Технологии разработки веб-приложений»

на тему: «Проектирование архитектуры и макетов веб-приложения»

Выполнил:

ст. гр. ИТ-123

Зеничева Э.С.

Принял:

Лексин А.Ю.

Владимир, 2025

**Цель работы**: Развитие навыков разработки и согласования технических спецификаций на веб-приложения.

**Задание**

Создать рабочее пространство для выполнения лабораторных работ и курсовой работы. Спроектировать и задокументировать информационную и программную архитектуру веб-приложения, разработать макеты страниц.

**Ход работы**:

Создадим Git репозиторий для выполнения лабораторных и курсовых работ в дальнейшем, как показано на рисунке 1.

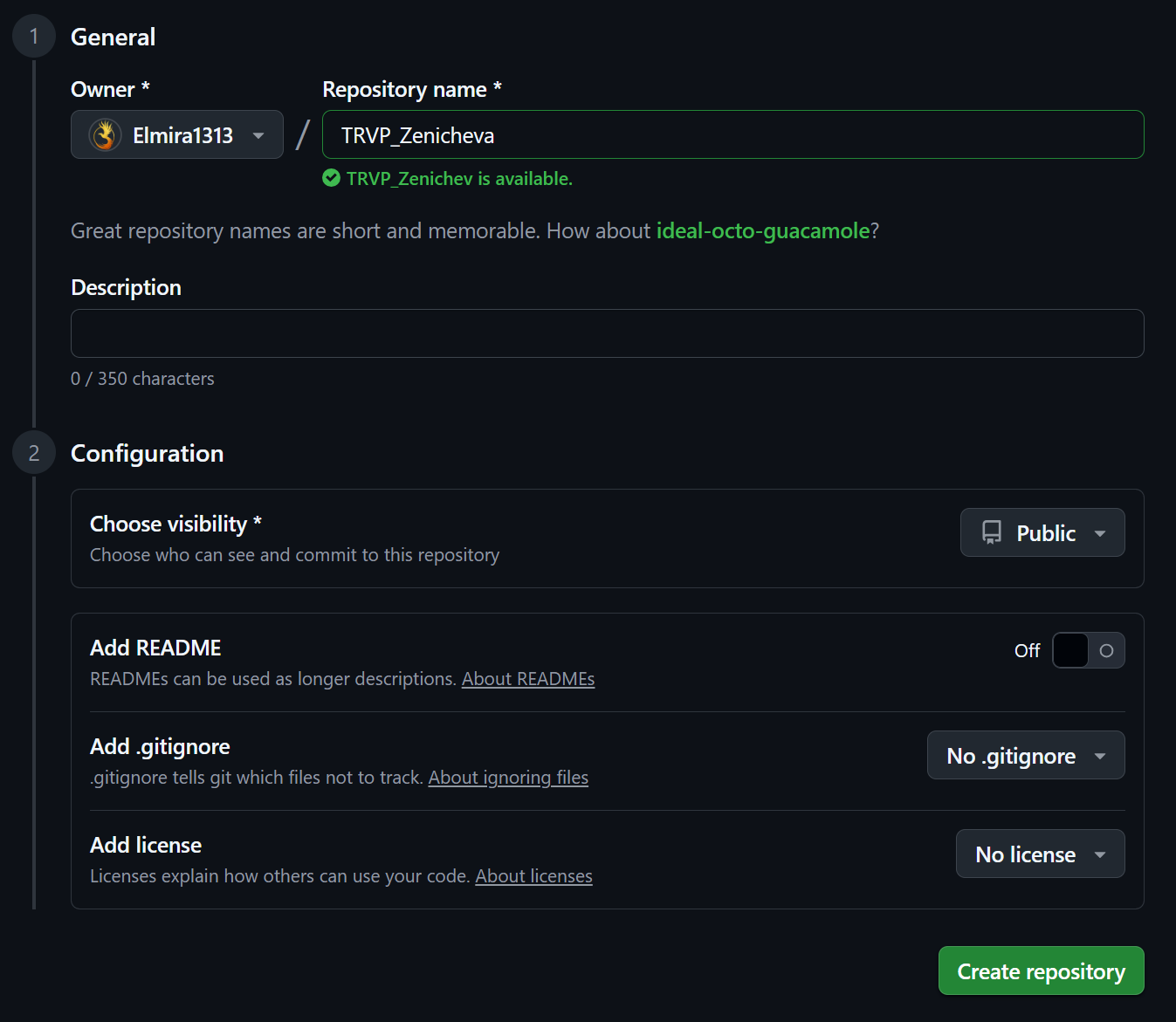


Рисунок − Создание репозитория

Описание предметной области.

Разрабатываемое приложение предназначено для учета (каталогизации) книг в домашней библиотеке, а также создания списка книг, которые пользователю хотелось бы приобрести. Его создание обусловлено тремя факторами:

* со временем личные библиотеки, которые собираются семьями или отдельными пользователями, становятся достаточно большими, и запомнить полный перечень книг становится сложно;
* пользователи, которые только начинают формировать библиотеку, часто имеют список желаемых книг, но его также неудобно хранить в памяти или записывать вручную;
* не все книги, не всех изданий можно найти в доступных, аналогичных ресурсах.

Отсюда следует, что в приложении можно будет записывать книги, которые уже есть в наличии, причем «расставлять» на разные полки или секции при желании, и книги, которые хотелось бы приобрести. Также в дальнейшем рассматривается возможность добавления друзей, для просмотра библиотек друг друга.

Функциональные требования.

В соответствии с предметной областью, участники могут зарегистрироваться и иметь личный кабинет, в котором в свою очередь можно создавать, редактировать и удалять свои библиотеки, секции, полки и книги на них.

Таким образом, основные сценарии пользователя:

* регистрация;
* вход;
* управление библиотекой;

Также рассматривается возможность взаимодействия разных пользователей посредством добавления в список друзей и просмотра аккаунтов друг друга.

Отсюда добавляются следующие сценарии:

* управление списком друзей;
* просмотр библиотек других пользователей;
* пометка книг с полки желаний другого пользователя;

Последний пункт предназначен для фиксации книг, которые можно приобрести в подарок для другого человека.

Общая диаграмма основных вариантов использования представлена на рисунке 2.

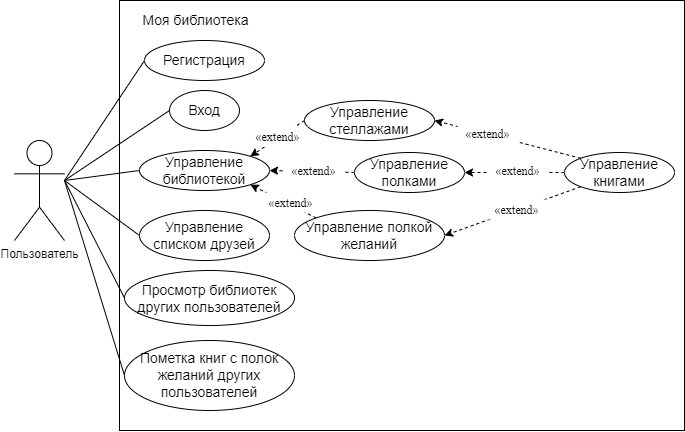


Рисунок − Диаграмма вариантов использования для пользователя

Структура сайта.

На основе описания предметной области и диаграммы вариантов использования, можно выделить основной функционал и на основе его продумать информационную структуру веб-сайта:

* Главная страница. На ней можно реализовать информационную страницу, на которой может быть краткое описание функционала приложения, с предложением войти в систему или зарегистрироваться, если аккаунта еще нет.
* Личный кабинет. Здесь можно реализовать возможность редактирования профиля, например, аватара, имени, статуса/описания, увлечений, возраста и т.д.
* Библиотеки. В этом разделе будет списком отображаться библиотеки пользователя. В приложении рассматривается наличие нескольких библиотек у одного аккаунта. Отсюда, соответственно, дальше можно попасть к списку секций (логически объединенные книги или полки книг) или полок (логически объединенных книг). Данные логические разделения не будут обязательными.
* Полка желаний. Здесь соответственно будут выводиться книги, которые пользователь хотел бы приобрести.
* Список друзей. Аналогично, вывод списка пользователей, находящихся в друзьях у данного аккаунта.
* На рисунке 3 представлена карта сайта, с описанными выше основными экранами, доступными пользователю.

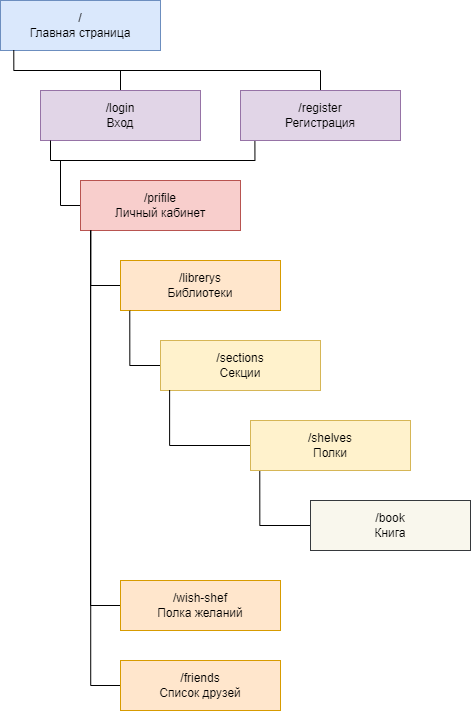


Рисунок − Карта сайта

Макеты веб-приложения.

На рисунках 4-8 представлены макеты следующих видов страниц:

* Макет главной страницы. Здесь будут располагаться приветственный блок, описание веб-ресурса, предложение входа и регистрации.
* Макет профиля. Содержит аватар, имя, другую личную информацию и настройки личного кабинета.
* Макет страницы со списком. Демонстрирует блоки пунктов того или иного списка (списка библиотек, секций, полок, друзей и т.д.).
* Макет страницы с книгами. Выводит на экран книги, так или иначе сгруппированные.
* Макет страницы книги. Выводит информацию о конкретной книге.

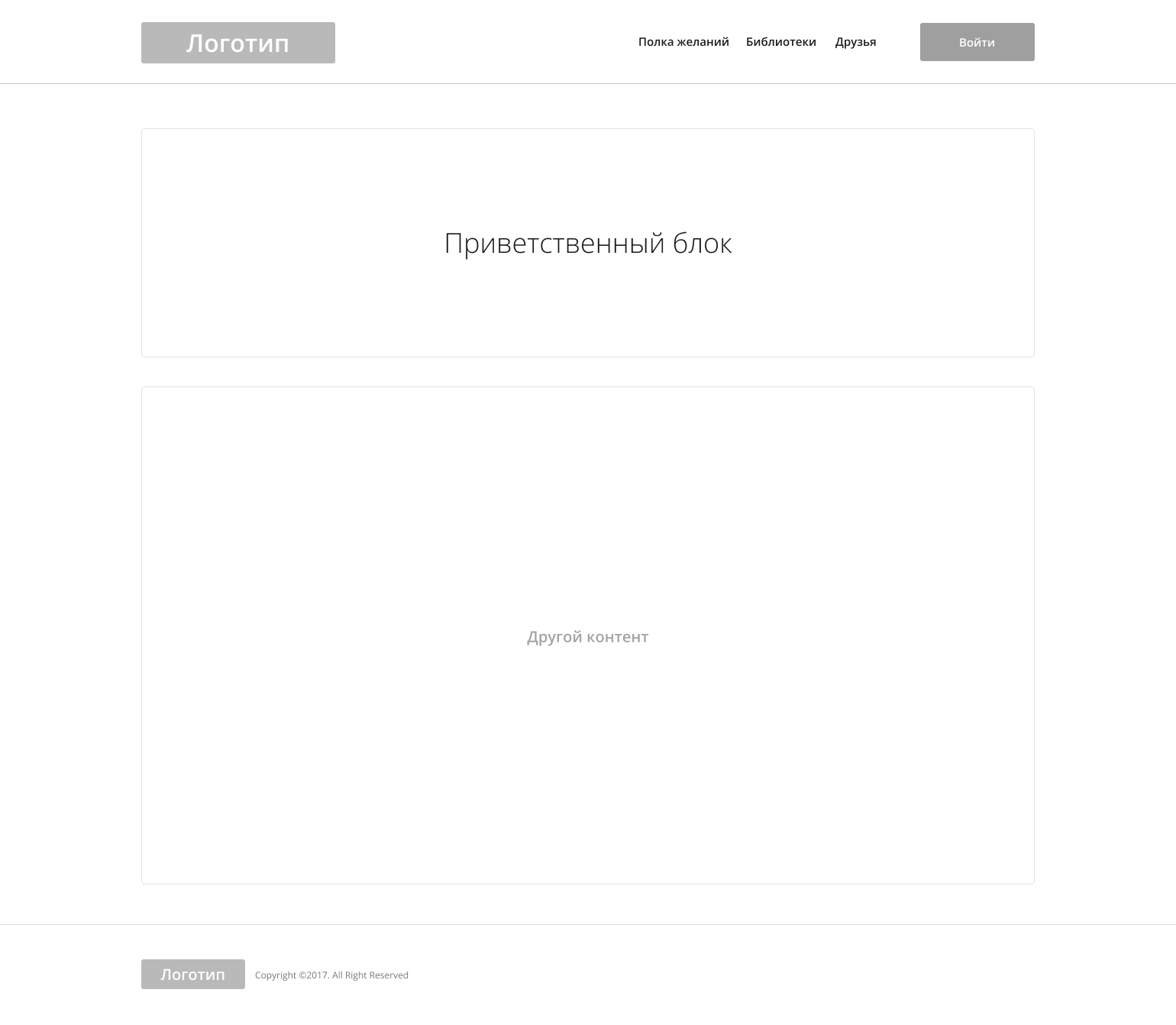


Рисунок − Макет главной страницы

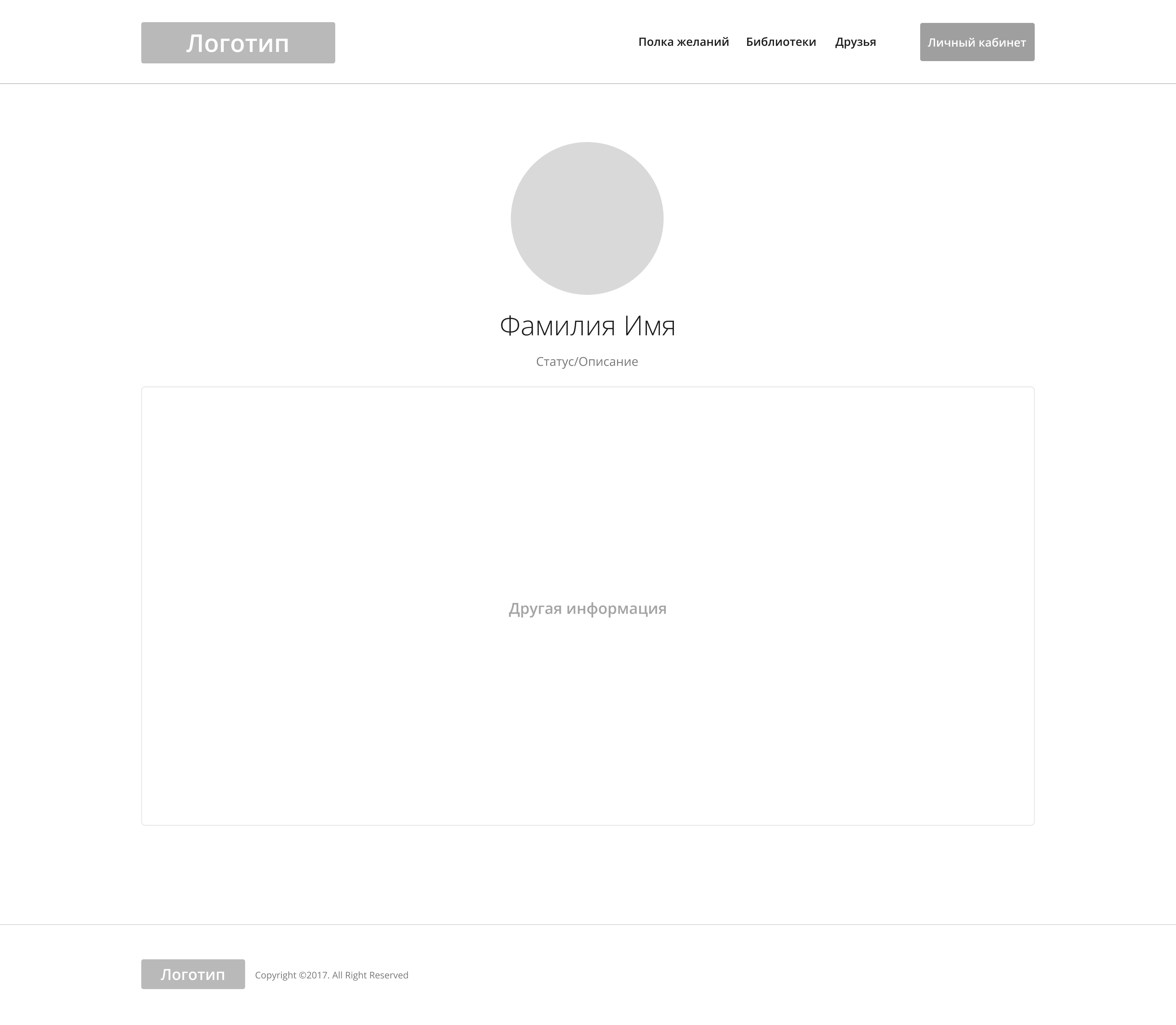


Рисунок − Макет профиля

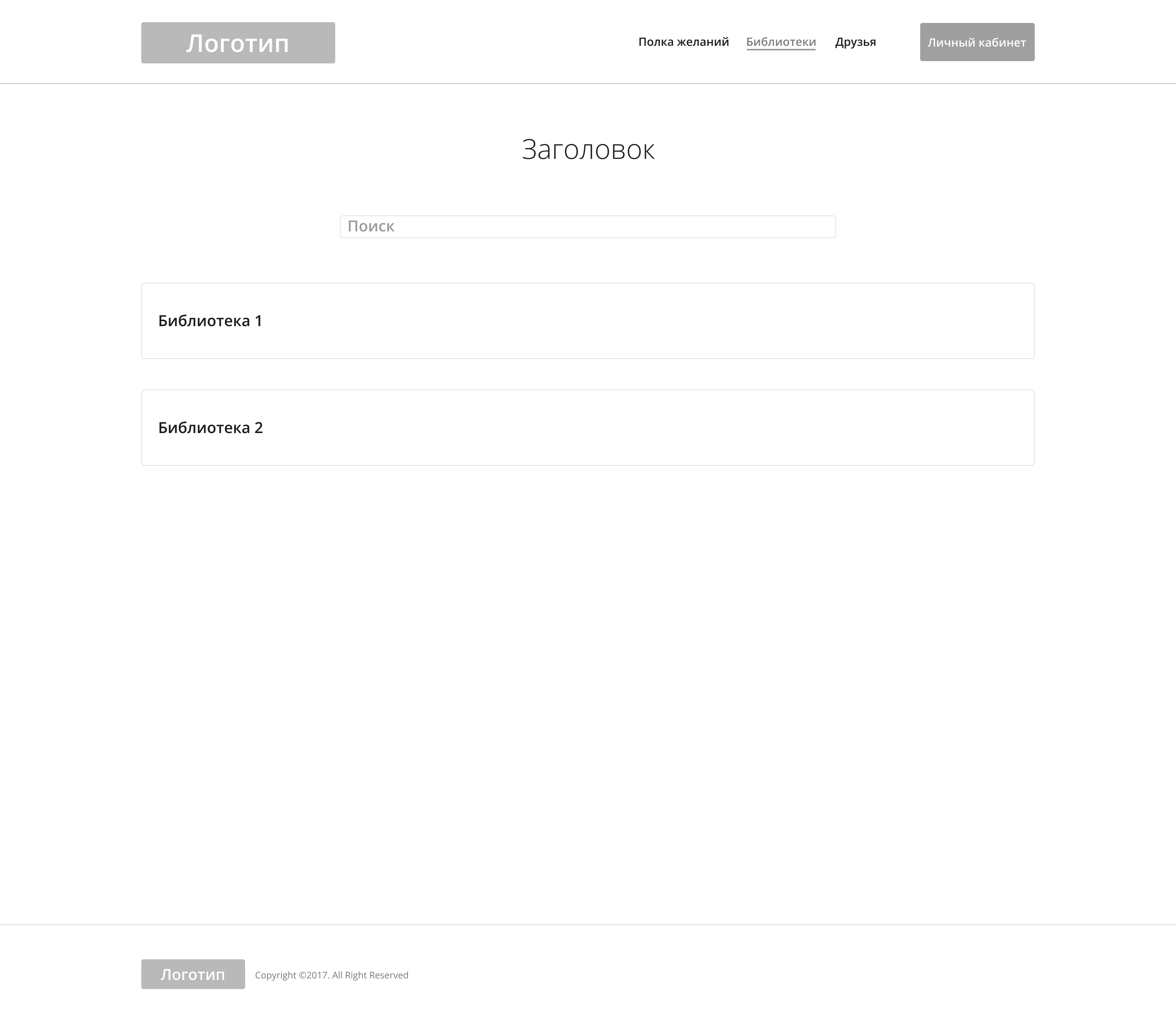


Рисунок − Макет страницы со списком



Рисунок − Макет страницы с книгами

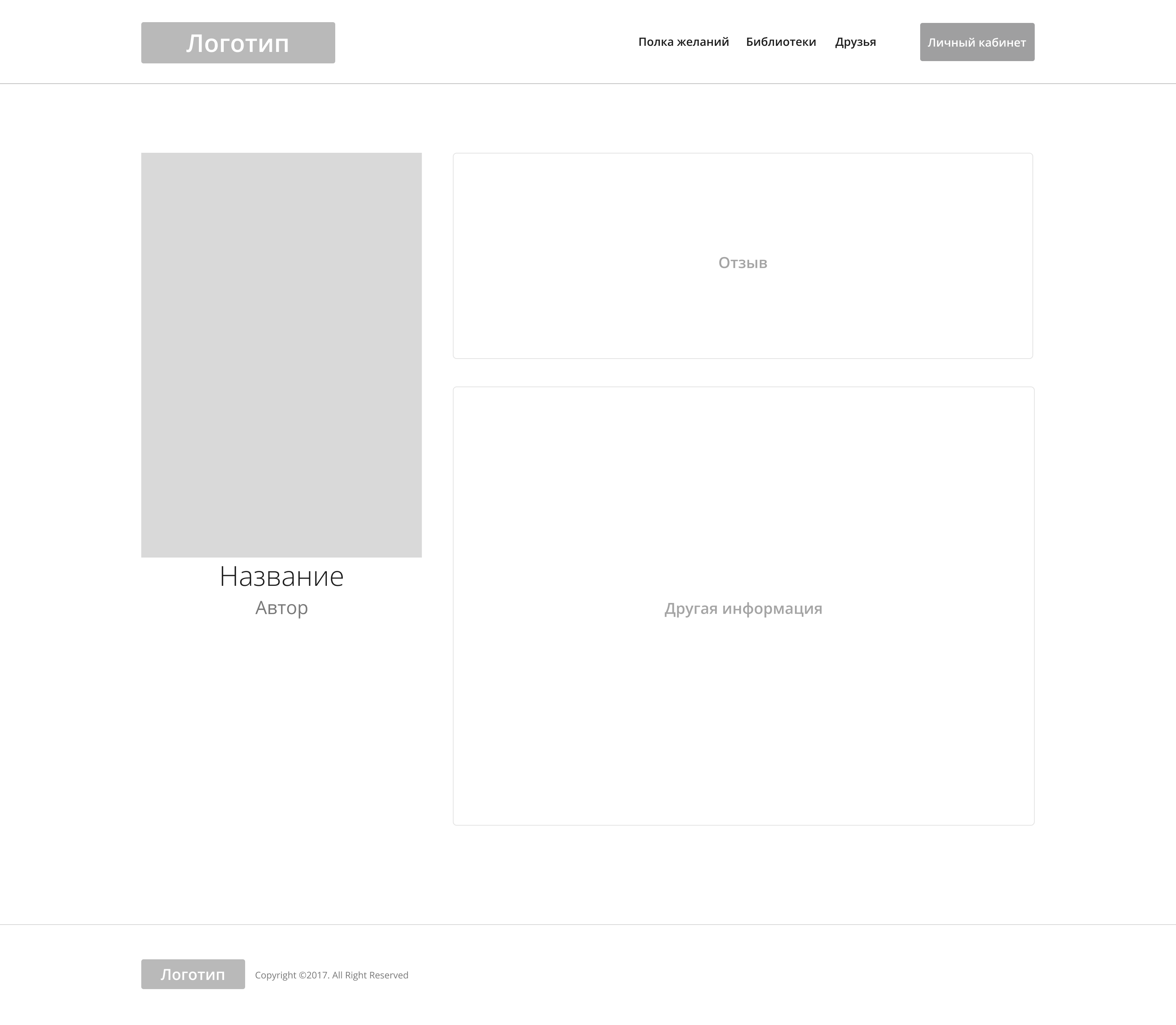


Рисунок 8 − Макет страницы книги

Архитектура веб-приложения.

Архитектура строится по классической многослойной модели.

Пользователь через браузер взаимодействует с клиентским слоем, реализованным через React SPA (Single Page Applications). Соответственно сайт планируется одностраничным, т.е. веб-приложение будет загружать всю необходимую разметку и скрипты при первом посещении и затем динамически обновлять контент в ответ на действия пользователя. Основные функции данного слоя:

* отображение интерфейса пользователя;
* управление состояниями (для перерисовки интерфейса);
* отправка запросов к серверу.

Клиентский слой (Frontend) в свою очередь в ответ на некоторые действия пользователя взаимодействует с серверным слоем (Backend) посредством rest api. API (Application Programming Interface) − «интерфейс для общения программ». REST (Representational State Transfer) − стиль проектирования API, основанный на использовании HTTP-запросов:

* GET - получить данные;
* POST - создать новый объект;
* PUT - изменить существующий объект;
* DELETE - удалить объект.

А Backend уже отправляет sql-запросы в базу данных, получает ответ и передает его по цепочке обратно.

Также возможно использование вспомогательных сервисов, таких как поиск, файловое хранилище или, например, уведомления.

Схема архитектуры веб-приложения показана на рисунке 8.

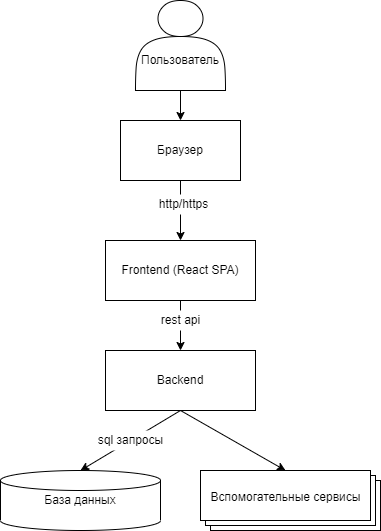


Рисунок − Программная архитектура приложения

Созданный ранее репозиторий с созданными спецификациями находится по ссылке: <https://github.com/Elmira1313/TRVP_Zenicheva.git>

**Вывод:** в ходе выполнения лабораторной работы было создано рабочее пространство для выполнения лабораторных работ и курсовой работы, а также было выполнено:

* описание предметной области;
* создание диаграммы вариантов использования и карты сайта;
* разработка основных макетов страниц приложения;
* проектирование программной архитектуры веб-приложения.