Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчёт

по предмету «Проектирование интернет систем»

Лабораторная работа №6

«Объектно-ориентированное модулирование. Физические диаграммы UML»

Студент: Козека Е. М.

ФИТ 4 курс 4 группа

Преподаватель: Парамонов А. И.

Минск 2025

**1 Постановка задач**

Программное средство должно быть выполнено с использованием асинхронного программирования, взаимодействовать с базой данных, реализовано под разными платформами. Программное средство должно представлять собой web-приложение с асинхронным UI. Отображение, бизнес логика и хранилище данных должны быть максимально независимы друг от друга для возможности расширения. Диаграмму вариантов использования разработать на основе UML, также необходимо разработать логическую схему базы данных и структурную схему приложения. Язык разработки проекта JavaScript, платформа Node.js. Развёртывание конечного приложения для последующего использования должно осуществляться с использованием Docker. Web-приложение должно быть логически завершенным. Управление программой должно быть интуитивно понятным и удобным. Листинги проекта должны содержать комментарии.

Web-приложение поддерживает 3 роли: гость, пользователь, администратор.

Функциональные требования для пользователя с ролью «Гость» включают только регистрацию и авторизацию.

Функциональные требования для пользователя с ролью «Пользователь»:

* просмотр главной страницы с каталогом домашних животных;
* поиск и фильтрация пород животных по виду, гипоаллергенности, продолжительности жизни, размеру;
* просмотр подробной информации о животном (изображение, вес, размер, продолжительность жизни, описание, гипоаллергенность, дрессируемость);
* просмотр списка своих питомцев и их карточек (кличка, аватар, день рождения, вес, состояние здоровья);
* управление карточками питомцев (добавление, редактирование, удаление);
* просмотр статей в сообществе;
* управление сохраненными статьями (добавление и удаление из избранного);
* добавление, редактирование и удаление своих статей;
* получение уведомлений о событиях по уходу за питомцами.

Функциональные требования для пользователя с ролью «Администратор» включают все возможности обычного пользователя, а также:

* добавление, редактирование и удаление видов животных;
* добавление, редактирование и удаление пород животных;
* добавление, редактирование и удаление тем статей;
* модерация заявок на добавление новых пород (одобрение или отклонение);
* удаление опубликованных пользователями статей.

Основные задачи приложения:

* разработать удобный интерфейс, который позволит пользователям получать полную информацию о животных;
* предоставить функционал для управления своим списком питомцев и получения пользователем напоминаний об уходе за ними;
* создать систему взаимодействия пользователей, где они смогут делиться советами по уходу за различными животными.

**2 Описание программных средств**

Draw.io (diagrams.net) — это современный и удобный сервис для построения различных схем и диаграмм, который стал популярным благодаря своей бесплатной основе и широким возможностям. Одним из ключевых преимуществ платформы является простота освоения: ею одинаково легко пользоваться как новичкам, так и опытным специалистам. Благодаря интуитивному интерфейсу создание блок-схем, UML-диаграмм, сетевых структур, бизнес-процессов, моделей IDEF0 или организационных схем не занимает много времени.

Инструмент оснащён богатой библиотекой шаблонов и стандартных элементов, что позволяет ускорить процесс разработки диаграмм. Пользователь может использовать готовые фигуры и иконки для разных задач, подстраивая их под свои проекты.

Сервис поддерживает экспорт в популярные форматы — PNG, JPG, SVG и PDF, что делает его универсальным для презентаций, печати или использования в документах. Для последующего редактирования проекты можно сохранить в формате XML, возвращаясь к ним при необходимости для внесения изменений. Дополнительно Draw.io легко интегрируется с облачными хранилищами вроде Google Drive, Dropbox и OneDrive, обеспечивая совместную работу, быстрый доступ с разных устройств и удобный обмен файлами.

Для работы в команде предусмотрена функция одновременного редактирования диаграмм несколькими пользователями и добавления комментариев. Плюсом является и наличие настольной версии программы, позволяющей работать офлайн без подключения к интернету.

Платформа также поддерживает интеграцию с корпоративными системами и инструментами управления проектами, включая Atlassian Confluence и Jira. Это делает её полезной для визуализации процессов и данных внутри больших проектов. Кроме того, предусмотрены импорт и экспорт файлов из других сервисов, таких как Lucidchart или Microsoft Visio, что облегчает переход с альтернативных решений.

В итоге Draw.io сочетает удобство, гибкость, поддержку множества форматов и тесную интеграцию с популярными сервисами, что делает его одним из лучших инструментов для визуализации информации. Платформа подходит как для индивидуальной работы, так и для совместных проектов, предоставляя все необходимые возможности для продуктивного создания диаграмм.

**3 Описание практического задания**

* 1. **Список компонент, из которых будет состоять подсистема**

1. Компонент React Components:

* Технология: React 19.2.0, JavaScript (ES6+);
* Описание: компоненты React-страницы.

1. Компонент React Pages:

* Технология: React 19.2.0, JavaScript (ES6+);
* Описание: страницы React, встраиваемые в SPA.

1. Компонент Axios Client:

* Технология: Axios 1.12.2, JavaScript;
* Описание: клиент для выполнения HTTP-запросов.

1. Компонент REST API:

* Технология: Express.js, Node.js;
* Описание: логический компонент, реализующий REST API для взаимодействия с клиентами.

1. Компонент Middleware:

* Технология: Express.js Middleware, Node.js;
* Описание: модуль промежуточной обработки запросов.

1. Компонент Services:

* Технология: Node.js, JavaScript;
* Описание: модуль бизнес-логики приложения.

1. Компонент Controllers:

* Технология: Node.js, JavaScript;
* Описание: обработки бизнес-логики запросов.

1. Компонент Routes:

* Технология: Express.js Router, Node.js;
* Описание: модуль определения эндпоинтов API.

**3.2 Связи между компонентами подсистемы**

**Ассоциация (Association):**

1. .env → index.js;
2. package.json → index.js;
3. .env → api.js;
4. package.json → api.js.

**Композиция (Composition):**

1. REST API → index.js;
2. Axios Client → api.js;
3. React Components → React Pages.

**Зависимость (Dependency):**

1. Services → schema.prisma;
2. Controllers → Services;
3. Controllers → uploads;
4. Routes → Controllers;
5. Routes → Middleware;
6. Middleware → uploads;
7. index.js → Routes;
8. docker-compose.yaml → index.js;
9. index.html (React SPA) → React Pages;
10. index.html (React SPA) → AuthContext;
11. AuthContext → api.js.

**3.3 Список физических устройств, на которых будет работать подсистема**

Клиентское устройство DESKTOP-3SMK9QM:

* + Процессор: минимум 2 ядра, 1.5+ GHz;
  + Оперативная память: минимум 2 GB RAM;
  + Хранилище: 100+ MB свободного места;
  + Операционная система: Windows 10 Pro;
  + Сетевое подключение: Интернет-соединение минимум 1 Мбит/с;
  + Браузер: Yandex Browser v25.10.1.1136 (64-bit).

Сервер приложения WINSERV-01:

* + Процессор: 8+ ядер, 2.0+ GHz;
  + Оперативная память: 8+ GB RAM;
  + Хранилище: 10+ GB свободного места;
  + Операционная система: Ubuntu 20.04 LTS, Windows Server 2022;
  + Сетевое подключение: стабильное интернет-соединение минимум 10 Мбит/с.

Сервер базы данных WINDB-XGHBH:

* + Процессор: 8+ ядер, 2.0+ GHz;
  + Оперативная память: 16+ GB RAM;
  + Хранилище: 50+ GB свободного места;
  + Операционная система: Ubuntu 20.04 LTS, Windows Server 2022;
  + Сетевое подключение: стабильное сетевое подключение к серверу приложения (минимум 1 Гбит/с в локальной сети).

**3.4 Протоколы связи между устройствами**

Между клиентским устройством и сервером приложения — HTTPS/1.1, между сервером приложения и сервером базы данных — TCP/IP.

**4 Диаграмма**

Диаграмма развертывания и компонентов представлена на рисунке 1.



Рисунок 1 — Диаграмма развертывания и компонентов