МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра анализа данных и искусственного интеллекта

Допустить к защите
заведующий кафедрой
д-р тех. наук, доцент
_____ А.В. Коваленко
_____2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)

РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ИГРОВОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

Работу выполнил(а) _____ М.И. Ковалев
(подпись)

Направление подготовки 09.03.03 Прикладная информатика

Направленность (профиль) Прикладная информатика в экономике

Научный руководитель
зав. каф., д-р. техн. наук, проф. _____ А.В. Коваленко
(подпись)

Нормоконтролер
канд. физ.-мат. наук, доц. _____Г.В. Калайдина
(подпись)

Краснодар

2024

РЕФЕРАТ

Выпускная квалификационная работа 60 с., 16 рис., 10 источников,3 приложения.

СВЕРТНОЧНЫЕ НЕЙРОННЫЕ СЕТИ, МИКРОСЕРВИСНАЯ АРХИТЕКТУРА, ГЕНЕРАЦИЯ ВИДЕО, КОНТЕЙНЕРИЗАЦИЯ, НЕРЕЛЯЦИОННАЯ БАЗА ДАННЫХ

Объектом исследования выпускной квалификационной работы являются разработка Web API, проработка пользовательского игрового опыта, создание архитектуры приложения, разработка и обучение нейронной сети, настройка и конфигурация окружения разработки и производства, интеграция с фронтенд-частью приложения. Весь этот комплекс задач направлен на создание игрового приложения, использующего нейросетевые технологии.

Цель работы: разработка серверной части игрового приложения с использованием нейросетевых технологий.

В результате выпускной квалификационной работы было создано Web API, которое было интегрированно с фронтенд приложением и использовало разработанную нейронную сеть, квалифицирующую зевание игрока.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современную цифровую эпоху игровые приложения становятся неотъемлемой частью повседневной жизни миллионов людей. С ростом спроса на высококачественный игровой контент возрастает и важность применения передовых технологий в их разработке. Нейросетевые технологии открывают новые горизонты в создании игровых приложений, делая их более интеллектуальными и адаптивными к пользовательскому опыту.

Зевание – это непроизвольный рефлекс, который может быть вызван не только усталостью, скукой или сонливостью, но и социальной или психологической передаваемостью, когда наблюдение за зевающим человеком или даже мысль о зевании могут спровоцировать этот рефлекс у других. Соревнование, в котором участникам приходится подавлять данный биологоческий рефлекс, неподдающийся контролю, является уникальным и увлекательным. Для определения зевоты, как и других изменений мимики, удобно использовать нейросетевые технологии.

Целью данного исследования является определение эффективных методов разработки и создание полноценной нейросетевой системы для онлайн-игры.

Актуальность темы исследования подчеркивается стремительным развитием области искусственного интеллекта и машинного обучения, которые лежат в основе нейросетевых технологий. Это исследование нацелено на анализ современных методик разработки игровых приложений с использованием нейросетей, а также на определение наиболее перспективных направлений для улучшения качества игрового процесса.

При разработке игровых приложений разработчики сталкиваются с выбором между множеством инструментов и подходов, каждый из которых предлагает различные возможности для интеграции искусственного интеллекта. Например, классификация эмоций человека с помощью камеры.

Важно не только создание технически продвинутых игровых механик, но и учет психологии пользователя, для чего применяются анализ поведения игрока и адаптивные системы обучения. Успешное игровое приложение должно быть не только технически совершенным, но и интересным для пользователя.

Данное исследование включает несколько ключевых разделов:

1) Обзор современных нейросетевых технологий и их применение в разработке игр.

2) Методология разработки игрового приложения с использованием нейросетевых технологий.

3) Разработка архитектуры нейросетевой модели и ее обучение.

4) Практическая реализация игрового приложения и его интеграция с нейросетевыми модулями.

Каждый из этих разделов предоставит ценные данные для разработчиков и исследователей, стремящихся улучшить качество и инновационность в области создания игр.

1 Теоретический анализ нейронных сетей и Web API приложений

1.1 Нейронные сети

Программное обеспечение, которое воссоздает структуру нейронных связей человека, известно как искусственная нейронная сеть. Такие программы применяются для разработки систем, обладающих способностью к обучению и способными осуществлять распознавание или создание контента [1].

В отличие от традиционного программного обеспечения, нейронная сеть требует прохождения процесса обучения, чтобы иметь возможность автономно справляться с заданиями. Это делает её поведение менее предсказуемым, однако при этом существенно расширяет её функциональность.

Благодаря таким возможностям высокопроизводительные нейронные сети находят применение в широком спектре задач, включая создание изобразительного искусства, сочинение поэзии и решение сложных проблем. Их использование простирается от виртуальных помощников до продвинутых систем диагностики в медицине [2].

В зависимости от характеристик нейронные сети поддаются различным классификациям:

1) Исходя из типа обучения:

–С учителем, где сеть корректирует свои параметры для минимизации ошибки, опираясь на предоставленные правильные ответы.

–Без учителя, где сеть самостоятельно выявляет закономерности в данных, не опираясь на предварительно заданные ответы.

2) Основываясь на архитектуре:

–Архитектура прямого распространения, где обработка информации происходит в одном направлении от входа к выходу без циклов. Пример такой сети представлен на рисунке 1.

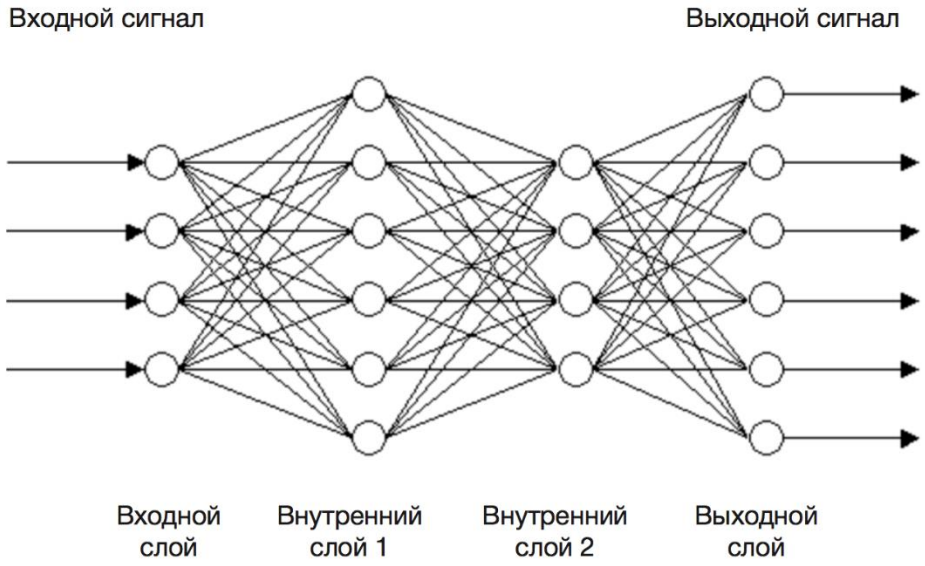


Рисунок 1 – Пример архитектуры прямой нейронной сети [3]

– Архитектура рекуррентной нейронной сети, где присутствуют обратные связи, позволяющие информации возвращаться в предыдущие слои, что делает такие сети идеальными для работы с последовательными данными, включая текст и временные ряды. Пример такой сети представлен на рисунке 2.

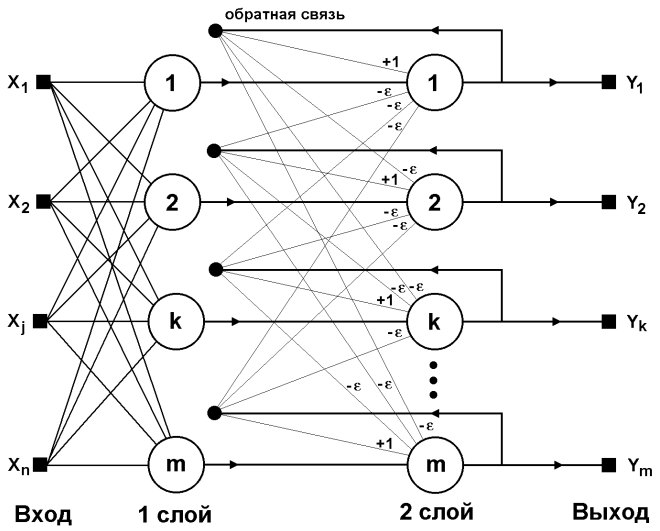


Рисунок 2 – Пример архитектуры рекуррентной нейронной сети [4]

В зависимости от количества и связности слоев, нейронные сети делятся на:

– Однослойные, представляющие собой простейшую форму нейронной сети с единственным слоем вычислений.

– Многослойные, состоящие из нескольких слоев, что позволяет проводить более глубокий и детальный анализ данных.

Кроме того, нейронные сети можно классифицировать по типу активационных функций:

– Линейные, где активационная функция передает сигнал без изменений, что полезно при линейных взаимосвязях в данных.

– Нелинейные, использующие функции активации для моделирования более сложных и многогранных отношений между входными данными.

Сам выбор конкретной архитектуры нейронной сети зависит от поставленной задачи:

а) Для классификации – задачи, где требуется распознавать объекты или относить их к определённым категориям, например, идентификация лиц или классификация изображений животных.

б) Для регрессии – задачи прогнозирования непрерывных значений, такие как оценка стоимости недвижимости или определение возраста по фотографии.

в) Для кластеризации – задачи группировки данных без заранее определённых категорий, например, сегментация рынка на основе поведения клиентов.

г) Для генерации – задачи создания нового контента, такого как тексты, изображения или видео.

1.2 Системы компьютерного зрения

Компьютерное зрение, опирающееся на нейронные сети, представляет собой одну из наиболее динамично развивающихся сфер в области искусственного интеллекта. Эти передовые технологии наделяют машины способностью интерпретировать и понимать визуальные данные из окружающего мира, подобно тому, как это делает человеческий мозг [5].

В центре нейросетевого компьютерного зрения находятся сверточные нейронные сети (CNN), специально разработанные для обработки данных с сеточной топологией, таких как изображения, где пространственные отношения между пикселями играют ключевую роль в их распознавании и классификации. Эти сети способны автоматически и эффективно выделять значимые признаки на различных уровнях сложности: от простых контуров и углов до сложных объектов, таких как лица или предметы. Обученные на огромных массивах аннотированных данных, CNN достигают высокой точности в задачах распознавания объектов, сегментации и классификации изображений.

Давайте рассмотрим примеры задач машинного зрения:

– Классификация изображений дает возможность компьютерам увидеть изображение и соотнести их к какому-то классу. Машинное зрение подразделяет изображения на типы и соответственно маркирует их, например автобусы, светофоры или собаки. Не надо далеко ходить в поисках примера реализации, камера современного смартфона может распознать лицо на фотографии и сфокусироваться на нем.

– Обнаружение объектов: задача компьютерного зрения по обнаружению и локализации изображений. Система пользуется классификацией для идентификации, сортировки и организации файлов изображения. В качестве примера рассмотрим, камеры для домов, которые используют функцию обнаружения объектов для обработки видео потоков с

камер в реальном времени, чтобы обнаруживать живые и не живые объекты в режиме реального времени и предоставлять информацию их владельцам.

– Сегментирование является алгоритмом машинного зрения, который определяет объект с помощью разделения изображения на области на основе видимых пикселей. Сегментирование выделяет главное в изображении, например, размещает форму или контур объекта и определяет, чем он является. Вдобавок сегментирование определяет наличие множества объектов на картинке. В качестве примера рассмотрим изображение, на котором есть тигр и лев, с помощью сегментирования можно распознать этих двух животных. Разница между обнаружением объекта и сегментированием состоит в том, что первое строит рамку для объекта, а второе отслеживает пиксели, чтобы определить форму объекта, а это, в свою очередь, упрощает анализ и маркировку изображения.

– Извлечение данных изображений по контенту – это использование методов компьютерного зрения, которые позволяют находить определенные изображения в базах данных. Данный пример изучает метаданные: описание, хештеги, метки. ключевые слова и т. д. Такой семантический поиск пользуется такими командами, как «изображения жирафов», для получения желаемого результата.

1.3 Веб API приложения

API – «Интерфейс прикладного программирования» является набором инструкций, которой дает возможность определенной программе взаимодействовать с другой. Простым языком: API связывает приложения с разным обеспечением или написанные на разных языках. В конечном счете такие программы формируют основу настоящего цифрового мира[6].

Одним из основных типов API считается Web. Он дает возможность разработчикам получать доступ к веб-сервисам при помощи протоколов,

таких как HTTP / HTTPS, с использованием которых, интеграция приложений с веб-инструментами значительно упрощается.

Отличием от других типов API является отсутствие состояния – данные или пользовательская информация не хранятся. Такая особенность дает возможность повторного использования без необходимости дополнительной установки или настройки. Сравнивая с API REST, веб-API использует большее количество протоколов: REST использует HTTP/HTTPS, в свою очередь веб-API помимо HTTP/HTTPS используют такие протоколы, как SOAP, BEEP, XML-RPC и JSON-RPC. Но есть и неудобства у Web, ведь он поддерживают только XML формат, когда тот же REST поддерживает еще и JSON.

Говоря о безопасности, в веб-API она зачастую более высокого уровня чем у REST API, так как происходит пользование большим количеством клиентов с разными целями, а REST наоборот почти всегда разрабатываются для внутренних пользователей поэтому и требуют меньшего количества протоколов безопасности. Но есть и обратная сторона такой безопасности: REST API обеспечивают более высокую производительность, чем web-API, так как имеют меньшее количество запросов и поддерживают HTTP-кэширование.

2 Технологии игрового приложения и нейронной сети

В главе обозреваются технологии, которые использовались для разработки приложения и нейронной сети.

2.1 Обзор технологий приложения

2.1.1 Язык программирования Java

Java – это язык программирования, первоначально разработанный Sun Microsystems и выпущенный в 1995 году как основная часть платформы Java. Этот язык заимствует многие части своей грамматики из C и C++, но имеет простую объектно-ориентированную модель и несколько низкоуровневых функций. Java при компиляции используется в форме байт-кода, который можно запускать на всех машинах моделирования Java, независимо от архитектуры и характеристик этого компьютера. Основные компоненты компиляторов Java, механизмы реализации и библиотеки публикуются этой компанией с 1995 года. В мае 1997 года компания предоставила бесплатное программное обеспечение для этого языка. Другие опубликовали другие реализации этого языка, такие как компилятор GNU для Java или OPENJDK. С появлением java2 новая версия смогла создавать новые комбинации для разных типов платформ. Например, J2EE, предназначенный для корпоративных приложений, и версия платформы Java, микроверсия, были выпущены для мобильных телефонов. В 1996 году в маркетинговых целях компания выпустила новую версию J2 под названиями Java Platform, Enterprise Edition, Java Platform, Micro Edition и Java Platform, Standard Edition. В 1997 году Sun Microsystems изменила организацию по стандартизации ISO/IEC JTC1 и Ecma International на формулу Java. Sun предоставила бесплатный доступ ко многим своим Java-приложениям. Компания Sun получала доход от продажи лицензий на некоторые из своих специальных приложений, таких как

Java Enterprise System. 13 ноября 1996 года Sun выпустила программное обеспечение Java бесплатно по общедоступной лицензии. Java широко используется для разработки приложений в ноутбуках, центрах обработки данных, игровых консолях, научных суперкомпьютерах, мобильных телефонах и т. д.[7].

Примеры использования Java:

– Он используется для разработки приложений для Android.
– Помогает вам создавать корпоративное программное обеспечение
– Широкий спектр мобильных Java-приложений
– Приложения для научных вычислений
– Использование для анализа больших данных
– Java-программирование аппаратных устройств
– Используется для серверных технологий, таких как Apache, JBoss, GlassFish и т. д.

Разница между тем, как работает Java и другие языки программирования, была революционной. Код на других языках сначала транслируется компилятором в инструкции для конкретного типа компьютера. Вместо этого компилятор Java преобразует код в нечто, называемое байт-кодом, который затем интерпретируется программным обеспечением, называемым Java Runtime Environment (JRE) или виртуальная машина Java . JRE действует как виртуальный компьютер, который интерпретирует байт-код и транслирует его для главного компьютера. По этой причине код Java может быть написан одинаково для многих платформ («напишите один раз, запустите где угодно»), что способствовало его популярности для использования в Интернете, где множество разных типов компьютеров могут получать одну и ту же веб-страницу.

2.1.1 Фреймворк Spring Boot

Sprint Boot – это Spring-фреймворк на основе Java, используемый для быстрой разработки приложений (для создания автономных микросервисов). Он имеет дополнительную поддержку автоматической настройки и встроенных серверов приложений, таких как Tomcat, Jetty и т. д.

Spring Boot предоставляет хорошую платформу для разработки автономных и промышленных приложений Spring для разработчиков Java, которые вы все еще можете запускать. Вы можете начать с минимальной конфигурации, не требуя полной настройки конфигурации Spring[8].

Особенности Spring Boot:

– Создает автономное Spring-приложение с минимальной необходимой настройкой.

– Он имеет встроенный Tomcat, Jetty, который позволяет просто писать код и запускать приложение.

– Предоставляйте готовые к использованию функции, такие как метрики, проверки работоспособности и внешняя конфигурация.

– Абсолютно нет требований к конфигурации XML.

Spring Boot автоматически настраивает ваше приложение на основе зависимостей, которые вы добавили в проект с помощью аннотации. Точкой входа приложения весенней загрузки является класс, содержащий аннотацию @SpringBootApplication и метод main.

2.2 Базы данных

2.2.1 PostgreSQL

PostgreSQL – это система, которая может управлять реляционными базами данных. Он похож на другие сетевые серверы баз данных SQL, такие как MySQL/MariaDB и Microsoft SQL, но есть некоторые ключевые различия

в функциях и функциях. PostgreSQL объектно-ориентирован, может хранить большое количество типов данных в столбцах (например, массивах) и поддерживает сложные запросы.

PostgreSQL может искать индексированные данные быстрее и более совместим со стандартами (см. стандарт совместимости данных ACID). PostgreSQL считается лучшим выбором по сравнению с другими решениями для баз данных при работе с гигантскими хранилищами данных (Microsoft даже использует его для отслеживания аналитики обновлений Windows с использованием базы данных, масштабируемой до петабайтов)

Библиотеки PostgreSQL существуют для всех основных сред программирования, включая Python, Javascript/Node.js, PHP и C++.

Некоторые популярные платформы, поддерживающие PostgreSQL, включают популярную среду PHP Laravel и среду Python Django. WordPress также поддерживает PostgreSQL с помощью плагина. Node.js, C++ и многие другие языки, платформы и платформы данных поддерживают PostgreSQL за счет использования соответствующей библиотеки для соответствующего языка программирования.

PostgreSQL имеет популярность среди специалистов по анализу данных, поскольку он поддерживает поля, содержащие массивы, и более сложные запросы, которые каскадируются между отношениями.

2.2.2 MinIO

MinIO – это самое быстрое объектное хранилище в мире, способное выполнять самый широкий набор рабочих нагрузок в отрасли.

Ориентируясь на высокую производительность, MinIO позволяет предприятиям поддерживать несколько сценариев использования на одной и той же платформе. Например, характеристики производительности MinIO означают, что вы можете выполнять несколько запросов Spark, Presto и Hive или быстро тестировать, обучать и развертывать алгоритмы искусственного

интеллекта, не сталкиваясь с узким местом в хранилище. Объектное хранилище MinIO используется в качестве основного хранилища для облачных приложений, которым требуется более высокая пропускная способность и меньшая задержка, чем может обеспечить традиционное объектное хранилище.

MinIO использует те же методы, что и гипермасштаберы, используя простоту в качестве строительного блока. Потребляя ресурсы небольшими дискретными порциями, MinIO использует дополнительные экземпляры и может плотно упаковывать их — вплоть до эксабайтного масштаба. Облачные приложения начинаются с небольшого размера на этапе прототипирования, но часто разрастаются до нескольких стоек, а иногда и до нескольких центров обработки данных, разбросанных по географическим регионам. MinIO предназначен для плавного масштабирования в соответствии с требованиями приложения.

MinIO разработан для мультитенантного мира Kubernetes. Благодаря MinIO арендаторы полностью изолированы друг от друга с помощью собственных экземпляров кластеров MinIO. Каждый арендатор, в свою очередь, может иметь несколько пользователей с разными уровнями привилегий доступа. Каждый клиентский кластер работает независимо друг от друга и благодаря размеру двоичного файла MinIO (~60 МБ) может быть плотно упакован для повышения эффективности. Могут использоваться стандартные балансировщики нагрузки HTTP или DNS с циклическим перебором.

2.2 Технологии создания нейронной сети распознавания психоэмоционального состояния человека на основе анализа выражения лица

Python – это язык программирования общего назначения, подходящий для разработчиков программного обеспечения, программистов и

специалистов по обработке данных. Он был создан в конце 1980-х и начале 1990-х годов разработчиком Гвидо ван Россумом. В начале 1991 года он выпустил первую версию языка Python. Есть несколько причин, по которым остается одним из наиболее популярных и востребованных языков программирования и занимает лидирующие позиции в различных сферах разработки продуктов, благодаря своим преимуществам:

1) Простота и читаемость кода: Ключевым атрибутом Python является его читабельность. Поскольку Python настолько близок к естественному языку пользователя и требует меньше строк кода, пользователь может быстро изучить и применить язык программирования к ряду операций. Независимо от того, являетесь ли вы новичком в программировании или опытным профессионалом, Python поможет вам сделать больше за меньшее время;

2) Адаптивность: Поскольку Python – такой универсальный язык — частично язык сценариев, частично интерпретируемый язык программирования — он используется во многих отраслях. Для всех, кто заинтересован в карьере в области технологий или обработки данных, Python – это самый близкий к универсальному магазину желанный навык. Работодатели из больших и малых компаний ищут людей, имеющих опыт работы с экосистемой Python;

3) Поддерживаемость: Статус Python с открытым исходным кодом поддерживается обширным сообществом пользователей . На своем веб-сайте Python хвастается, что его пользовательская база «полна энтузиазма и стремится к широкому распространению использования языка». Благодаря поддержке сообщества для всех уровней опыта база знаний Python продолжает расти. Вы можете найти ресурсы и советы сообщества Python в Slack и Discord, а также в специально подобранном еженедельном

4) Стандартные и сторонние библиотеки: Ученые, работающие с данными, могут принести пользу только в том случае, если данные, которые они анализируют, являются чистыми и полными и могут быть

проанализированы на более глубоком уровне. Python постоянно добавляет новые библиотеки, что дает ученым, работающим с данными, новые возможности для исследования и анализа данных инновационными способами, а также предоставляет предприятиям большую ценность и более значимую информацию. Например, модуль Scikit-learn предлагает набор инструментов машинного обучения, которые позволяют ученым прогнозировать результаты и разрабатывать алгоритмы на основе этих прогнозов. ионном бюллетене и в социальных сетях, включая Twitter, Facebook и IRC.

Для построения модели, определяющей зевание человека, нужны определенные библиотеки:

1) Keras – библиотека для построения моделей, по которым проходит обучение. Она подбирает нужное количество слоев, требуемого для точности.

2) TensorFlow – библиотека, выполняющая все низкоуровневые вычисления и преобразования и служит двигателем в создании модели нейронной сети.

3) OpenCV – библиотека, которая распознает изображение. Она использует компьютерное зрение, чтобы дать точную информацию, что изображено на картинке.

4) NumPy – библиотека, предоставляющая функции для выполнения математических функций над массивами.

5) Pillow – библиотека, работающая над форматами. Позволяет открывать, обрабатывать и сохранять изображения.

Особенности Keras

Keras предлагает простой и интуитивно понятный интерфейс для создания моделей нейронных сетей. С его помощью можно легко задавать слои, настраивать параметры и компилировать модель всего несколькими строками кода. Keras позволяет комбинировать различные слои, такие как полносвязные, сверточные и рекуррентные, упрощая создание сложных архитектур нейронных сетей. Она поддерживает различные бэкэнды, включая

TensorFlow, Theano и Microsoft Cognitive Toolkit (CNTK), что позволяет выбрать подходящий вариант в зависимости от предпочтений и требований проекта.

Keras подходит для выполнения разнообразных задач машинного обучения, таких как классификация, регрессия, сегментация изображений и обработка естественного языка. Библиотека включает множество встроенных функций для работы с данными и оценки моделей. Keras обладает гибкой архитектурой, позволяя добавлять пользовательские слои, функции потерь и метрики. Также она предоставляет доступ к предобученным моделям, таким как VGG16, ResNet и Inception, что позволяет быстро реализовывать задачи компьютерного зрения.

Sequential API – это самый простой способ использовать Keras для построения нейронной сети. Используя этот Sequentialкласс, можно объединять различные типы слоев один за другим для создания нейронной сети. В Keras Sequential API доступно множество типов слоев. Одним из наиболее распространенных типов слоев является Dense слой, полносвязный слой, но есть и множество других:

Сверточный слой: слой для обработки изображений — используется для сверточных нейронных сетей.

Рекуррентный слой: слой для обработки последовательностей данных – используется для рекуррентных нейронных сетей.

Слой MaxPooling: слой для понижения выборки карт объектов путем взятия максимального значения в неперекрывающихся прямоугольных блоках — используется для сохранения важных функций и одновременного уменьшения вероятности переобучения.

Сглаживающий слой: слой, который сглаживает многомерные входные тензоры в одно измерение — используется в качестве переходного слоя между сверточными или рекуррентными слоями и полностью связанными слоями в нейронной сети.

Слой исключения: слой, который случайным образом устанавливает входные единицы на 0 (с использованием определенной частоты) во время обучения – используется в качестве метода регуляризации для предотвращения переобучения в нейронных сетях.

Слой внедрения: слой, который представляет слова или фразы в многомерном векторном пространстве, используемый для сопоставления слов или фраз с плотными векторами для использования в качестве входных данных для нейронной сети.

Это всего лишь несколько примеров множества типов слоев, доступных в Keras Sequential API. Каждый уровень предназначен для выполнения определенного типа вычислений на входах, и их можно комбинировать для создания мощных архитектур нейронных сетей(рис. 4).

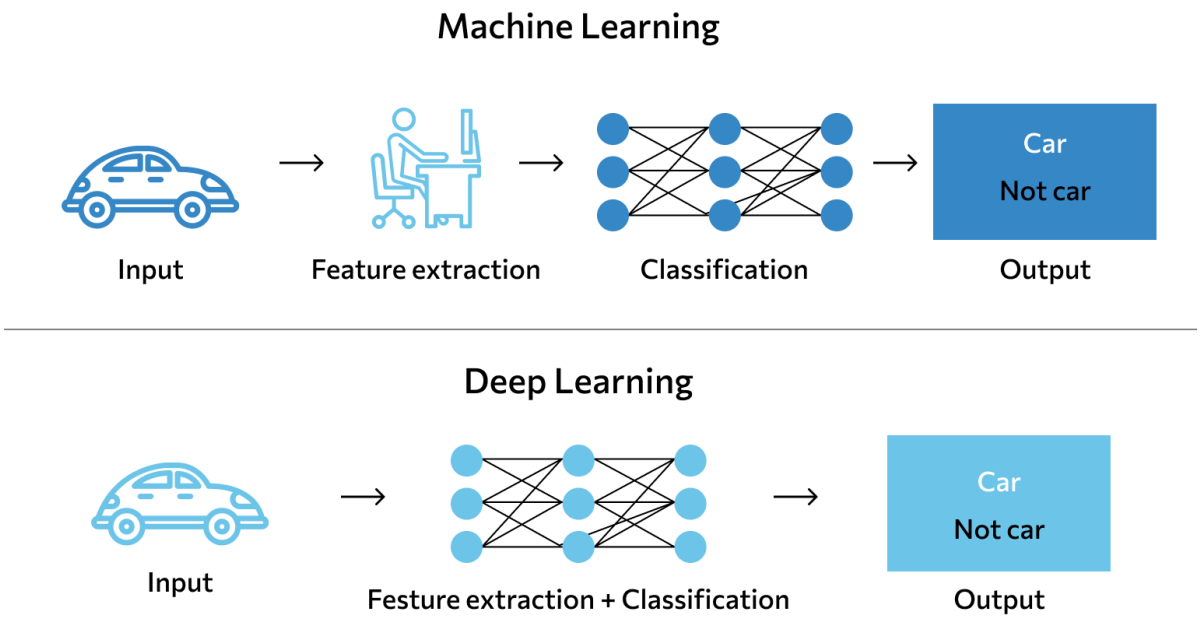


Рисунок 4 – Строение Keras

Чтобы создать собственный слой Keras, вы создаете класс R6, производный от KerasLayer. Существует три метода реализации (только один из них call()требуется для всех типов слоев):

– build(input_shape): здесь вы определите свой вес. Обратите внимание: если ваш слой не определяет обучаемые веса, вам не нужно реализовывать этот метод.

– call(x): здесь находится логика слоя. Если вы не хотите, чтобы ваш слой поддерживал маскирование, вам нужно заботиться только о первом аргументе, передаваемом в вызов: входном тензоре.

– compute_output_shape(input_shape): если ваш слой изменяет форму входных данных, вам следует указать здесь логику преобразования формы. Это позволяет Keras выполнять автоматический вывод формы. Если вы не измените форму ввода, вам не нужно реализовывать этот метод.

Помимо создания пользовательских слоев, вы также можете создать собственную модель. Это может быть необходимо, если вы хотите использовать активное выполнение TensorFlow в сочетании с обязательно записанным прямым проходом.

В тех случаях, когда в этом нет необходимости, но требуется гибкость в построении архитектуры, рекомендуется просто придерживаться функционального API.

Пользовательская модель определяется путем keras_model_custom () вызова функции, которая определяет создаваемые слои и операции, которые необходимо выполнить при прямом проходе.

TensorFlow – это платформа машинного обучения с открытым исходным кодом, разработанная Google, которая предоставляет мощный набор инструментов для ученых и разработчиков данных для создания, развертывания и обучения моделей машинного обучения. Первоначально он был выпущен в 2015 году как ранняя версия программного обеспечения, но с годами продолжал развиваться, предлагая более мощные возможности.

Библиотека TensorFlow позволяет разработчикам создавать сложные нейронные сети, используя различные языки программирования, такие как Python и JavaScript. Кроме того, TensorFlow упрощает развертывание моделей

на мобильных устройствах или облачных платформах, таких как Google Cloud Platform (GCP) и Amazon Web Services (AWS).

TensorFlow чаще всего используется для приложений глубокого обучения, таких как обработка естественного языка (NLP), распознавание изображений, классификация текста, обнаружение объектов, системы рекомендаций и многое другое.

Тензор – это массив данных, которые может обрабатывать TensorFlow . Тензор можно представить в виде матрицы или вектора. Проще говоря, мы можем думать об этом как о наборе чисел, расположенных в определенной форме. Математически тензор можно использовать для создания n-мерных наборов данных.

Нульмерный тензор – это скаляр, который содержит одно значение и не имеет осей. Одномерный тензор — это вектор, который содержит список значений и имеет одну ось. Двумерный тензор – это матрица, содержащая значения, хранящиеся по двум осям(рис. 5).

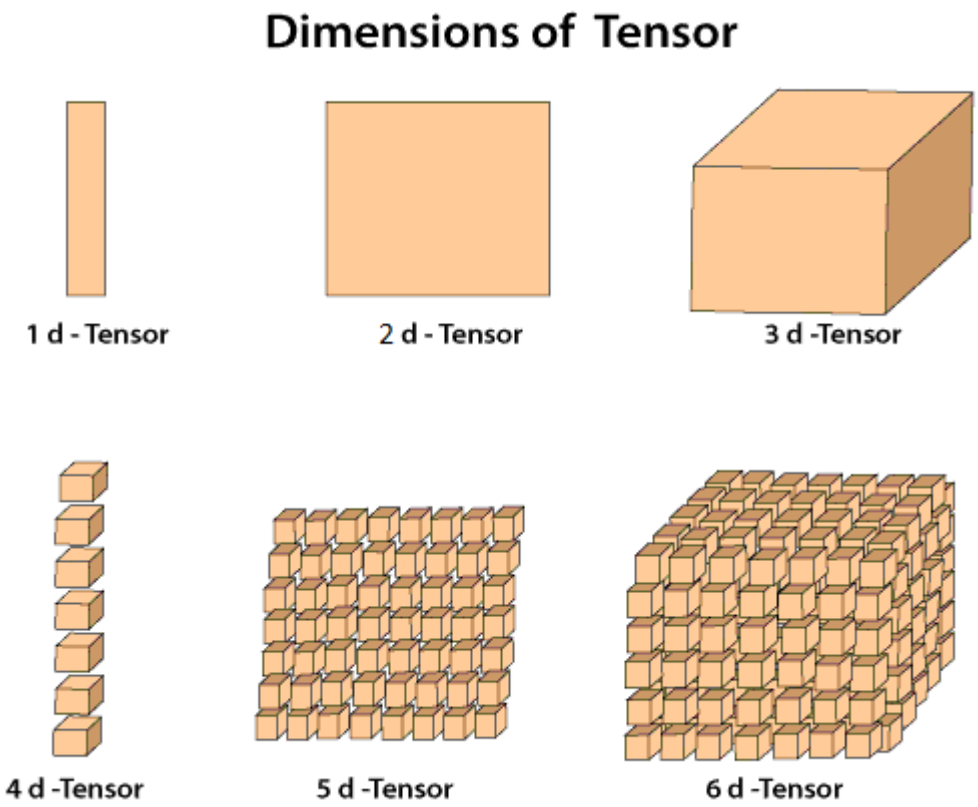


Рисунок 5 – Тензоры

TensorFlow используется в различных приложениях: от обработки естественного языка (NLP) и распознавания изображений до прогнозной аналитики и автономного управления транспортными средствами. Его можно использовать для обучения глубоких нейронных сетей обнаружению и классификации объектов, генерации рекомендаций, классификации изображений и создания голосовых приложений.

Кроме того, TensorFlow используется для прогнозирования, текстовых приложений, алгоритмической торговли и оптимизации. Он также используется в различных приложениях здравоохранения, таких как медицинская диагностика и разработка лекарств.

Все эти варианты использования демонстрируют универсальность и мощь, которые TensorFlow предлагает ученым и разработчикам данных. Благодаря своей гибкости и масштабируемости TensorFlow позволяет решать сложные задачи машинного обучения за гораздо меньшее время по сравнению с традиционными методами.

В основе TensorFlow лежит граф потока данных, который описывает, как данные перемещаются посредством серии операций или преобразований. Основная идея графа потока данных заключается в том, что операции выражаются в виде узлов, причем каждый узел выполняет одну операцию со своими входными данными. Входные и выходные данные операций передаются через ребра (тензоры). Это позволяет разбить сложные вычисления на более мелкие и более управляемые фрагменты.

TensorFlow также предоставляет ряд инструментов для построения и обучения нейронных сетей. Одним из самых популярных инструментов является утилита tf.keras, которая позволяет пользователям быстро создавать и обучать модели глубокого обучения без необходимости писать код с нуля. Он также включает в себя мощные инструменты визуализации, которые помогут пользователям понять данные и параметры модели.

TensorFlow также является расширяемым и может использоваться с различными языками программирования, включая Python, C++, JavaScript и

Go. Он также поддерживает работу на графических процессорах (графических процессорах) для максимальной производительности. TensorFlow благодаря своему мощному набору функций является одной из самых популярных платформ для глубокого обучения и продолжает развиваться по мере разработки более мощных алгоритмов.

Особенности TensorFlow

– В TensorFlow модели представлены в виде графов, которые являются математическими абстракциями, состоящими из вершин и путей между ними. Граф можно сравнить со схемой дорог, соединяющих разные точки. В программировании графы полезны для решения задач маршрутизации и создания нейронных сетей.

– TensorFlow оперирует тензорами, многомерными структурами данных в направленном пространстве, используемыми в линейной алгебре и физике. Название библиотеки происходит от этих тензоров, которые описывают пути графа, в то время как вершины представляют математические операции.

– В TensorFlow вычисления описываются как потоки данных через граф, где информация «движется» по путям от одной вершины к другой.

– Библиотека может работать как на обычных центральных процессорах (CPU), так и на графических процессорах (GPU), с возможностью переключения режимов в коде. Также существует тензорный процессор TPU, разработанный специально для TensorFlow, доступный через облачные сервисы Google.

Преимущества:

1) TensorFlow позволяет сосредоточиться на логике программы и математике, не беспокоясь о технических деталях реализации. Это упрощает разработку, предоставляя TensorFlow заботу о вычислениях.

2) TensorFlow позволяет создавать модели поэтапно и проверять отдельные компоненты, что удобнее, чем описывать граф целиком. Такой

подход обеспечивает гибкость и возможность интерактивной настройки структуры модели.

3) TensorFlow подходит для создания нейронных сетей, глубокого обучения и других задач машинного обучения. Благодаря графам и тензорам, можно легко моделировать сложные математические структуры. TensorFlow работает как с центральными, так и с графическими процессорами и легко интегрируется с другими инструментами, такими как Keras.

4) TensorFlow работает на популярных операционных системах, локально и в облаке. Есть расширения для мобильных устройств (TensorFlow Lite), IoT и браузеров (TensorFlow.js для JavaScript и Node.js).

5) TensorFlow имеет широкую пользовательскую базу, что облегчает поиск ответов на вопросы. Сообщество активно развивает библиотеку, создаёт новые продукты и дополнения, пишет документацию и обучающие материалы, что упрощает начало работы и максимизирует полезность TensorFlow.

Недостатки:

а) TensorFlow, разработанный Google, придерживается уникальных стандартов компании. Изначально предназначенная для внутреннего использования, библиотека может демонстрировать неочевидное поведение, усложняющее отладку кода. Эти сложности можно преодолеть, внимательно изучая документацию и используя дополнительные инструменты для отладки.

б) При использовании с графическим процессором TensorFlow захватывает всю его память, что может снизить производительность. Если используется несколько моделей на разных фреймворках, TensorFlow может вызвать ошибки, занимая всю видеопамять. Для предотвращения этого потребление памяти следует ограничивать вручную.

в) Машинное обучение само по себе сложно, и TensorFlow с его специфичными стандартами может быть непростой для новичков библиотекой.

Структура OpenCV:

Сейчас структура OpenCV – это множественные модули для разных целей:

а) хранения математических функций и вычислений, алгебры и структур данных;

б) хранения моделей для машинного обучения;

в) ввода и вывода картинок или видео, чтения и записи в файл;

г) обработки изображения;

д) распознавания примитивов;

е) детектирования объектов – лиц, предметов и других;

ж) отслеживания и анализа движений на видео;

з) обработки трехмерной информации;

и) ускорения работы библиотеки;

к) хранения устаревшего или еще не готового кода и других.

Каждый модуль узко специализирован. Их не нужно скачивать отдельно: в пакет установки включена вся основная функциональность библиотеки.

OpenCV позволяет выполнять различные операции с изображением:

– Чтение изображения: OpenCV помогает вам прочитать изображение из файла или непосредственно с камеры, чтобы сделать его доступным для дальнейшей обработки.

– Улучшение изображения: вы сможете улучшить изображение, отрегулировав яркость, резкость или сжатие изображения. Это полезно для визуализации качества изображения.

– Обнаружение объектов: Как вы можете видеть на изображении ниже, объект также можно обнаружить с помощью OpenCV, браслета, часов, узоров и лиц. Это также может включать в себя распознавание лиц, форм и даже объектов.

– Фильтрация изображений. Вы можете изменить изображение, применив различные фильтры, такие как размытие или повышение резкости.

– Рисование изображений: OpenCV позволяет рисовать текст, линии и любые фигуры на изображениях.

– Сохранение измененных изображений: после обработки вы можете сохранить изменяемые изображения для будущего анализа.

Существует множество приложений, которые решаются с использованием OpenCV, некоторые из них перечислены ниже:

1) Распознавание лица

2) Автоматизированный контроль и наблюдение

3) количество людей – подсчитать (пешеходность в торговом центре и т. д.)

4) Транспортные средства рассчитывают движение по автомагистралям и их скорость

5) Интерактивные арт-инсталляции.

6) Выявление аномалий (дефектов) в производственном процессе (нечетная бракованная продукция)

7) Сшивка изображений просмотра улиц

8) Поиск и извлечение видео/изображений

9) Навигация и управление автомобилем без участия робота и водителя

10) Распознавание объектов

11) Анализ медицинских изображений

12) Фильмы – 3D-структура из движения

13) Распознавание рекламы телеканалов

Преимущества:

1) OpenCv используется на многих языках программирования по всему миру, в том числе в Google и Microsoft. Из-за этого библиотека имеет большую аудиторию. Поэтому документация на нее есть на многих языках, в том числе и русский.

2) У OpenCV бесплатный доступ. Любой желающий может скачать лицензию и посмотреть исходный код библиотеки, чтобы узнать, как реализована та или иная функция.

3) OpenCv включает в себя чуть менее 3000 алгоритмов компьютерного зрения, что позволяет решать сложные задачи по распознаванию изображения.

4) Библиотека работает на быстрых скоростях нежели Matlab, поэтому ей проще распознать или обработать картинку.

5) Из-за быстрой скорости OpenCv можно использовать в реальном времени. Поэтому моя выпускная квалификационная работа сделана с помощью ее. Веб-камера приложения работает в реальном времени и считывает биометрику лица пользователя, определяя зевание. Без OpenCv это сделать невозможно.

Недостатки:

а) Чтобы пользоваться OpenCv нужны знания машинного обучения

б) У библиотеки отсутствуют коды обработки ошибок. Если она возникла, то сложно понять, где именно.

в) OpenCv работает на больших платформах, если ее запустить на обычном компьютере, то будет маленькая производительность.

Существует несколько форматов изображений, с которыми вы можете работать с помощью модуля Python Pillow. Вероятно, вы наиболее знакомы с такими форматами растровых изображений, как JPG, PNG и GIF и другими.

Растровые изображения имеют фиксированное количество пикселей в зависимости от разрешения изображения, и каждый пиксель имеет определенный цвет. Если вы увеличите растровое изображение достаточно сильно, пиксели станут более заметными. Подавляющее большинство изображений хранится таким образом.

С другой стороны, в векторных изображениях для создания изображений используются кривые, определяемые математическими уравнениями. Вы можете продолжать масштабировать векторное

изображение, и кривые останутся плавными. Двумя примерами этого формата файлов являются SVG и EPS.

Однако работа с векторными изображениями в Python может быть немного сложной, поскольку требует использования других специализированных библиотек.

Сильной стороной модуля Python Pillow является его полезность при изменении изображений. Включено множество функций предварительной и постобработки изображений.

NumPy работает с многомерными массивами. Как и OpenCv, работу выполняет быстро. У нее есть расширения – SciPy, которое использовалось в выпускной квалификационной работе.

Все эти библиотеки нужны для разработки систем искусственного интеллекта для распознавания психо – эмоционального состояния человека на основе анализа выражения лица, с целью предоставления пользователю инструмента для мониторинга и анализа своих эмоциональных состояний в реальном времени.

2.3 Docker

Docker – это технология виртуализации с открытым исходным кодом, которая упрощает создание, тестирование и развертывание приложений. С помощью Docker вы можете поставлять свои приложения в контейнерную среду, в которой находится все необходимое для запуска вашего приложения: от библиотек до системных инструментов, файлов конфигурации, кодов, зависимостей и среды выполнения[9].

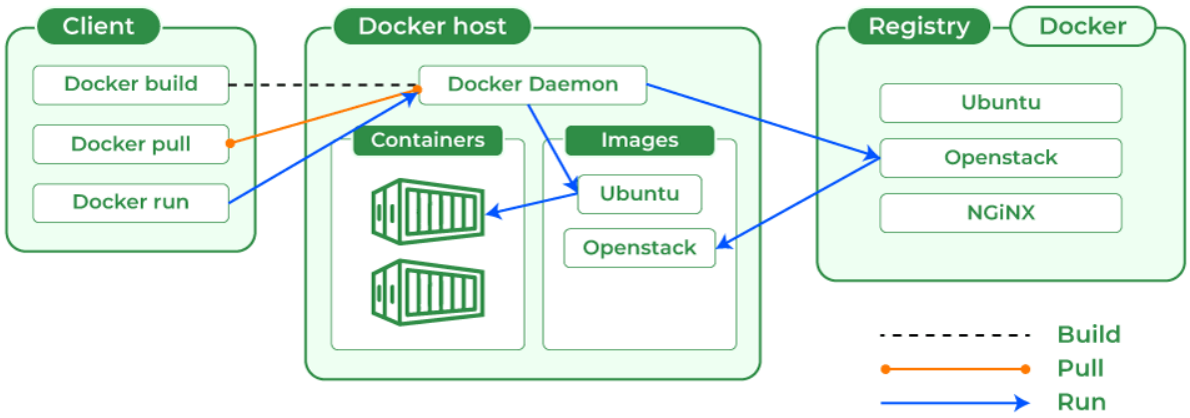


Рисунок 6 – Архитектура Docker

Docker использует технологии виртуализации и контейнеров. Используя эту технологию, Docker действует как изолированная песочница для создания легких контейнеров, упрощающих разработку и развертывание приложений.

Хотя это звучит очень похоже на виртуальные машины, они сильно отличаются. Виртуальные машины используют гипервизоры для совместного использования ресурсов и виртуализации аппаратных компонентов. В результате виртуальные машины используют ресурсы неэффективно, хотя консолидация серверов является большим преимуществом. С другой стороны, Docker виртуализирует операционную систему (ОС). Контейнеры работают на одном ядре и используют функции ОС, такие как группы управления (cgroups), для распределения доступных ресурсов между процессами Docker. Кроме того, доступ каждого процесса к ресурсам ограничен пространством имен для эффективного совместного использования ресурсов внутри контейнеров Docker.

3 Реализация игрового приложения с использованием нейросетевых технологий

Главе состоит из описания разработки всех частей приложения: база данных, нейросетевая модель «Зевок» и бэкенд часть приложения Yawn.

3.1 Проектирование базы данных

Были созданы 5 таблиц(рис. 7):

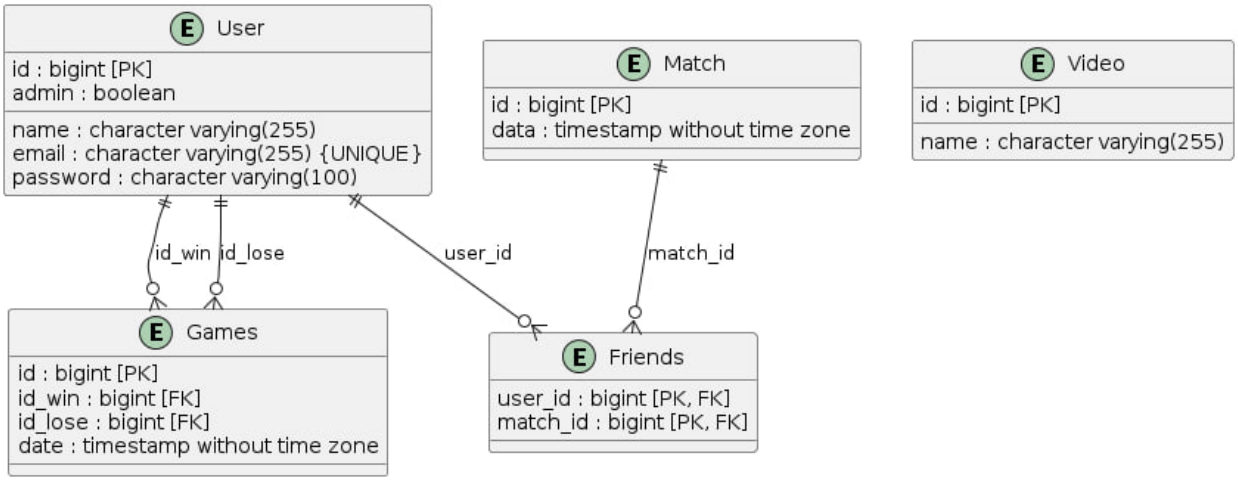


Рисунок 7 – ER диаграмма базы данных

Описание приведенных выше таблиц:

1) Таблица user:

{

а) id: идентификатор пользователя.

б) name: имя пользователя.

в) email: электронная почта пользователя.

г) admin: флаг, указывающий, является ли пользователь администратором.

д) password: пароль пользователя.

};

– CONSTRAINT constraint_email_user UNIQUE (email): ограничение, гарантирующее уникальность значения поля email.

2) Таблица match:

{

а) id: идентификатор мэтча.

б) date: дата мэтча.

};

3) Таблица friends:

{

а) user_id: идентификатор пользователя, который является другом.

б) match_id: идентификатор мэтча, в рамках которого пользователь является другом.

};

– PRIMARY KEY (user_id, match_id): составной первичный ключ, состоящий из идентификатора пользователя и идентификатора мэтча.

– FOREIGN KEY (match_id): внешний ключ, связывающий поле match_id с полем id таблицы match. При удалении матча из таблицы match также удаляются соответствующие записи в таблице friends.

4) Таблица games:

{

а) id: идентификатор игры.

б) id_win: идентификатор пользователя, победителя игры.

в) id_lose: идентификатор пользователя, проигравшего игру.

г) date: дата игры.

};

5) Таблица video:

{

– id: идентификатор видеофайла.

– name: имя видеофайла.

}

PRIMARY KEY (id): для всех таблиц, в котором он есть является первичным ключом.

3.2 Создание модели для определения зевания человека

3.2.1 Формирование dataset

Датасет Emotion Detection был взят из Kaggle[10] и был доработан таким образом, чтобы помимо представленных эмоций были добавлены изображения зевания. Итоговый датасет составил 35,685 элементов и был разделен на обучающую и тестовую выборку по 26685 и 9000 соответственно. Распознаваемые эмоции: счастье, нейтральность, печаль, гнев, зевота, отвращение и страх. Фрагмент датасета представлен на рисунке 8.

Рисунок 8 – Фрагмент датасета

3.2.2 Реализация создания и обучения нейросетевой модели

1) В самом начале создается пустая последовательная модель с помощью *Sequential()*. В эту модель будут последовательно добавлены слои.

2) Первый добавляемый слой - это *Conv2D* слой с 64 фильтрами размером (3, 3), функцией активации ReLU и указанием размера входных данных *(48, 48, 1)* (48 пикселей в высоту, 48 пикселей в ширину, 1 канал, поскольку изображения предполагаются в оттенках серого). Слой выполняет операцию свертки с использованием указанных фильтров.

3) Затем следует *BatchNormalization* слой, который нормализует выходы предыдущего слоя, ускоряя обучение.

4) После этого добавляется *MaxPooling2D* слой с размером пула (2, 2), который уменьшает размерность выхода предыдущего слоя, выбирая максимальное значение в каждом окне пула.

5) Чтобы предотвратить переобучение, используется *Dropout* слой с коэффициентом отсева 0.25, который случайным образом обнуляет 25% элементов входа.

6) Далее добавляются *Conv2D* слой с 128 фильтрами размером (5, 5) и *Conv2D* слой с 512 фильтрами размером (3, 3), после каждого из которых следуют *BatchNormalization*, *MaxPooling2D* и *Dropout*.

7) Затем следует слой *Flatten,* который преобразует выходы предыдущего слоя в одномерный вектор, чтобы его можно было подать на полносвязные слои.

8) Последующие три слоя являются полносвязными слоями. У первых двух слоев 256 и 512 нейронов соответственно, и функция активации для обоих слоев - ReLU.

9) Каждый полносвязный слой сопровождается *BatchNormalization* слоем для нормализации выходов и *Dropout* слоем с коэффициентом отсева 0.25.

10) Заключительный слой *Dense* содержит 7 нейронов (поскольку в задаче предполагается классификация на 7 классов) и использует функцию активации softmax для получения вероятностного распределения по классам.

Затем для обучения модели используется оптимизатор Adam со скоростью обучения 0.0005 и функцией потерь категориальная перекрестная энтропия.

После обучения нейронной сети на 50 эпохах ошибка и точность распознавания составили: 0.231 и 91%, 0.492 и 88% – для обучающей и тестовой выборки соответственно.

Архитектура описанной выше нейронной сети представлена на рисунке 9.

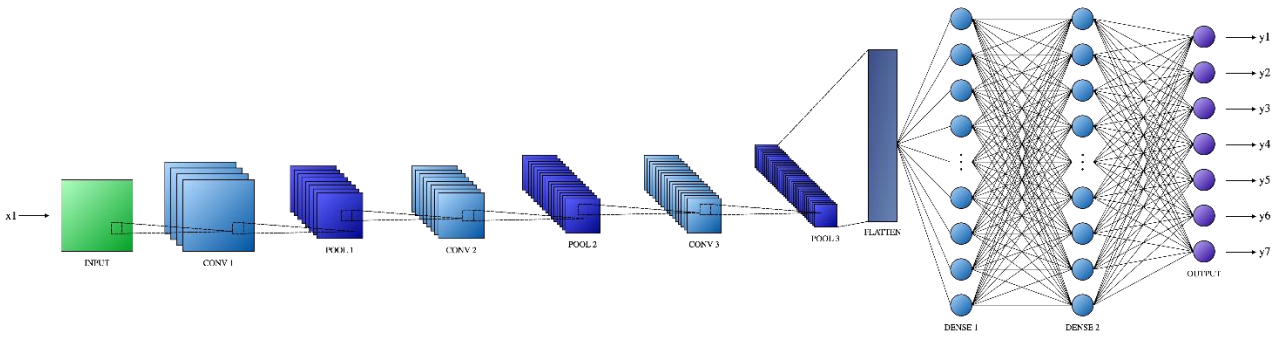


Рисунок 9 – Архитектура модели

– X – это входные данные, в виде изображений формата 48x48 серых оттенков.

– Y(1..7) являются выходными коэффициентами распознавания эмоций:

1) счастье.
2) нейтральность.
3) печаль.
4) гнев.
5) зевота.

6) отвращение.

7) страх.

– Сумма всех Y равна 1.

3.3 Разработка бэкэнд части приложения Yawn

Созданное приложение состоит представленных на рисунке 10 компонентов:

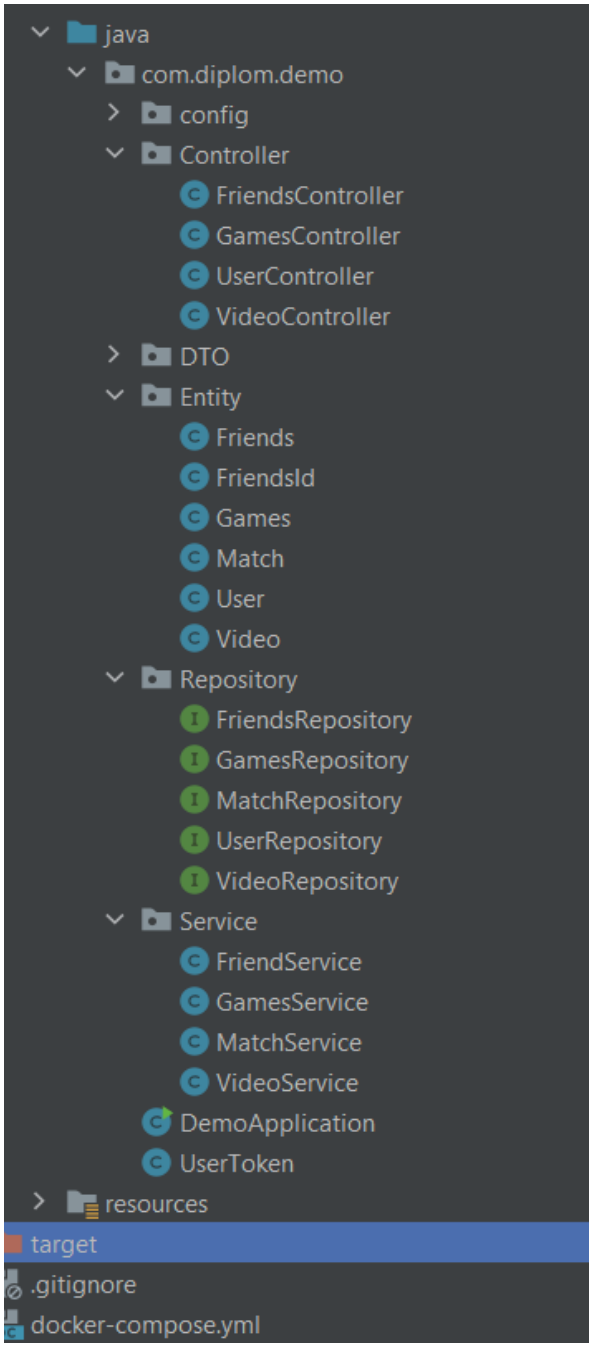


Рисунок 10 – Компоненты разработанного приложения

3.3.1 Создание сущностей

Для начала были разработаны сущности:

1) Класс User – это сущность пользователя в системе, она включает в себя приведенные ниже поля:

– id:

а) Тип: Long

б) Описание: Уникальный идентификатор пользователя.

в) Связь с базой данных: автоматически генерируется при помощи GenerationType.SEQUENCE. Используется аннотация @GeneratedValue для указания стратегии генерации и настройки последовательности.

– name:

а) Тип: String

б) Описание: Имя пользователя.

– email:

а) Тип: String

б) Описание: Электронная почта пользователя.

в) Ограничения: Должен быть уникальным.

г) Связь с базой данных: Сохраняется в столбце email. Используется аннотация @Column(unique = true) для указания уникальности.

– admin:

а) Тип: Boolean

б) Описание: Признак администратора. true, если пользователь является администратором, и false в противном случае.

– password:

а) Тип: String

б) Описание: Пароль пользователя.

2) Класс Match: сущность мэтча в системе и включает в себя:

– id:

а) Тип: Long

б) Связь с базой данных: Как и в таблице User генерируется с помощью GenerationType.SEQUENCE, пользуясь @GeneratedValue.

– data: имеет тип LocalDateTime

3) Класс Friends: хранит информацию о дружбе между пользователями и содержит два поля, которые вместе формируют составной первичный ключ:

– userId:

а) Тип: Long

б) Описание: Идентификатор пользователя, который добавлен в друзья.

– matchId:

а) Тип: Long

б) Описание: Идентификатор матча, который связывает друзей.

4) Вспомогательный класс FriendsId используется для обозначения составного ключа. Он также содержит два поля: userId и matchId, которые должны совпадать по типу и названию с полями в классе Friends:

– userId:

а) Тип: Long

б) Описание: Идентификатор пользователя, который добавлен в друзья.

в) Связь с базой данных: Используется как часть составного ключа для класса Friends.

– matchId:

а) Тип: Long

б) Описание: Идентификатор матча, который связывает друзей.

в) Связь с базой данных: Используется как часть составного ключа Friends.

5) Games – сущность, которая хранит данные об играх ме. Она включает в себя идентификатор игры, победителя, проигравшего, а также дату и время проведения игры:

– id:

а) Тип: Long

б) Описание: Уникальный идентификатор игры.

в) Связь с базой данных: Аннотирован как первичный ключ с автоинкрементом. Поле генерируется с помощью последовательности games_sequence.

– id_win:

а) Тип: Long

б) Описание: Идентификатор пользователя, который выиграл игру.

в) Связь с базой данных: Поле для хранения информации о победителе игры.

– id_lose:

а) Тип: Long

б) Описание: Идентификатор пользователя, который проиграл игру.

в) Связь с базой данных: Поле для хранения информации о проигравшем игре.

– date:

а) Тип: LocalDateTime

б) Описание: Дата и время проведения игры.

в) Связь с базой данных: Поле для хранения информации о времени проведения игры.

6) Video является сущностью для хранения информации о видео:

– id:

а) Тип: Long

б) Описание: Уникальный идентификатор видео.

в) Связь с базой данных: Аннотировано как первичный ключ. Это поле служит для уникальной идентификации каждого видео в базе данных.

– name:

а) Тип: String

б) Описание: Название видео.

в) Связь с базой данных: Поле для хранения названия видео. Может использоваться для отображения названия видео в пользовательском интерфейсе или для поиска видео по названию.

3.3.2 Репозитории

Для обращения к базе данных было создано 5 репозиториев.

3.3.2.1 MatchRepository

В MatchRepository находятся поддерживаемые стандартные методы Spring Data JPA для выполнения основных операций CRUD с сущностями Match, такие как fingById, save, delete и т.д.

3.3.2.2 UserRepository

По мимо поддерживаемых стандартных методов Spring Data JPA, были созданы методы для вывода пользователей, управляемой через сущность User:

Методы:

1) findByEmail(String email): Находит пользователя по его email, возвращает объектов User.

2) findUsersWithNameStartingWith(Long userId, String namePrefix): Находит всех пользователей, чьи имена начинаются с заданного префикса и чьи ID не совпадают с указанным, возвращает список объектов User.

3) findFriendsById(Long userId, Long count): Находит указанное число друзей пользователя, возвращает список объектов User.

4) findAllFriendsById(Long userId): Находит всех друзей пользователя по ID, возвращает список объектов User.

5) findFriendByName(Long userId, String nick): Выполняет те же функции, что и findUsersWithNameStartingWith, только не для всех пользователей, а только для друзей.

6) findFriendById(Long userId, Long friendId): Находит по ID друга идентификатор мэтча и возвращает его, тип: Long.

3.3.2.3 FriendsRepository

По мимо сгенерированных методов Spring Data JPA, был создан метод для проверки дружбы в системе, управляемой через сущность Friends:

– isUsersFriends(Long userId, Long friendId): Проверяет, являются ли два пользователя друзьями, возвращает булевское значение (true или false), указывающее, являются ли два пользователя друзьями. Включает в себя подзапрос для проверки, если существует совпадение match_id между друзьями.

3.3.2.4 GamesRepository

По мимо поддерживаемых стандартных методов Spring Data JPA, были созданы методы для подсчета игр и побед пользователя в системе, а также вывод данных игры, управляемой через сущность Games:

1) findGamesByUserId(Long userId): Возвращает список игр, в которых указанный пользователь учувствовал.

2) findCountGamesByUserId(Long userId): Возвращает количество игр, в которых указанный пользователь принимал участие.

3) findCountWinByUserId(Long userId): Возвращает количество побед указанного пользователя.

3.3.2.5 VideoRepository

По мимо сгенерированных Spring Data JPA, был создан метод, обеспечивающий гибкость и возможность получения случайных данных о видеофайлах из базы данных:

– findRandomNames(): Метод возвращает список строк, представляющих случайно выбранные имена видеофайлов.

3.3.3 Реализация бизнес-логики

Для обработки бизнес-логики было создано 4 сервиса. Они выполняют различные операции над данными (создание, удаление и т. д.).

3.3.3.1 MatchService

MatchService создает и удаляет мэтчи в приложении, предоставляя методы для выполнения соответствующих операций с базой данных.

Методы:

– createMatch(): Создает и возвращает новый объект Match.

– deleteMatch(Long id)`:Метод удаляет матч с указанным идентификатором из базы данных. Использует метод deleteById() репозитория для удаления матча по его идентификатору.

3.3.3.2 FriendService

FriendService – это сервис был создан для добавления, просмотра и удаления друзей пользователя. Сервис пользуется помощью FriendsRepository, MatchService и UserRepository.

Реализованные Методы:

1) createFriendsForUsersWithMatchId(Long id, Long friendId) создает записи о друзьях для двух пользователей с заданными идентификаторами и связывает их с новым матчем:

– Создается новый матч с помощью matchService.createMatch().

– Для каждого пользователя создается объект Friends, устанавливается его идентификатор (i`) и идентификатор созданного матча (match.getId()).

– Оба объекта Friends сохраняются в базу данных с помощью friendsRepository.save().

2) deleteFriend(Long id, Long friendId) удаляет связь дружбы между двумя пользователями:

– Получает идентификатор матча, связанного с пользователями, из репозитория пользователей.

– Затем вызывает метод matchService.deleteMatch() для удаления матча с помощью его идентификатора.

3.3.3.3 GamesService

GamesService предназначен для работы с играми. Класс имеет зависимость от GamesRepository.

Методы:

1) createGame(Long winId, Long loseId) создает новую игру между двумя пользователями:

– Создается новый объект Games, в который записываются идентификаторы победителя (winId) и проигравшего (loseId), а также текущая дата и время.

– Объект Games сохраняется в базу данных с помощью gamesRepository.save().

2) gamesInfo(Long id) возвращает информацию о количестве игр и побед пользователя с заданным идентификатором.

– Создается новый объект UserGamesInfo, в который записываются количество игр, количество побед и коэффициент выигрышей (победы/игры).

– Информация извлекается из репозитория с помощью методов findCountGamesByUserId(id) и findCountWinByUserId(id) и вычисляется коэффициент выигрышей.

– Объект UserGamesInfo возвращается в качестве результата метода.

3.3.3.4 Видео сервис

VideoService – это сервис генерации видеороликов. Сервис имеет методы для получения, загрузки и объединения видеофайлов.

Методы:

1) downloadVideo(String bucketName, String objectName): Сохраняет видеофайл из указанного ведра в MinIO во временный файл;

2) getVideo(String bucketName, String objectName): Возвращает видеофайл из MinIO в виде InputStream;

3) generateFileList(List<Path> paths): Создает временный файл со списком видеофайлов.

4) concatenateVideos(Path ffmpegPath, List<String> videoNames, String outputPath): По полученному списку от generateFileList объединяет видеоролики;

5) mergeVideos(): Использует concatenateVideos и удаляет с помощью deleteFile временные файлы. Этот метод передается контроллеру;

1) getPathList(List<String> videoNames): Использует downloadVideo и возвращает список путей к временным файлам.

2) deleteFile(String filePath): Удаляет файл по указанному пути.

3.3.4 Реализация контроллеров

Были созданы контроллеры для обработки HTTP-запросов, которые используют методы сервисов и репозитории:

3.3.4.1 UserController

Контроллер для работы с пользователями предоставляет методы для создания, получения и изменения пользовательских данных. Для функционирования методы UserController используют UserRepository, который был описан выше.

Методы:

– createUser(String authorizationHeader): Создает нового пользователя на основе зашифрованных данных переданных в заголовке авторизации.

– getUserById(String authorizationHeader, Long id): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха возвращает данные пользователя по идентификатору. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getInfoUserById(String authorizationHeader): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха возвращает информацию о текущем пользователе на основе данных из заголовка. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getUserByName(String authorizationHeader, String namePrefix): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха возвращает список пользователей по начальным буквам имени из параметра namePrefix. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– updateUser(String authorizationHeader): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха обновляет данные текущего пользователя на основе данных из заголовка. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

В приложении А представлена диаграмма последовательностей для каждого из методов UserController. Данная диаграмма отражает передачу данных между разными компонентами приложения.

На рисунке 11 представлен результат выполнения http запроса на создание пользователя.

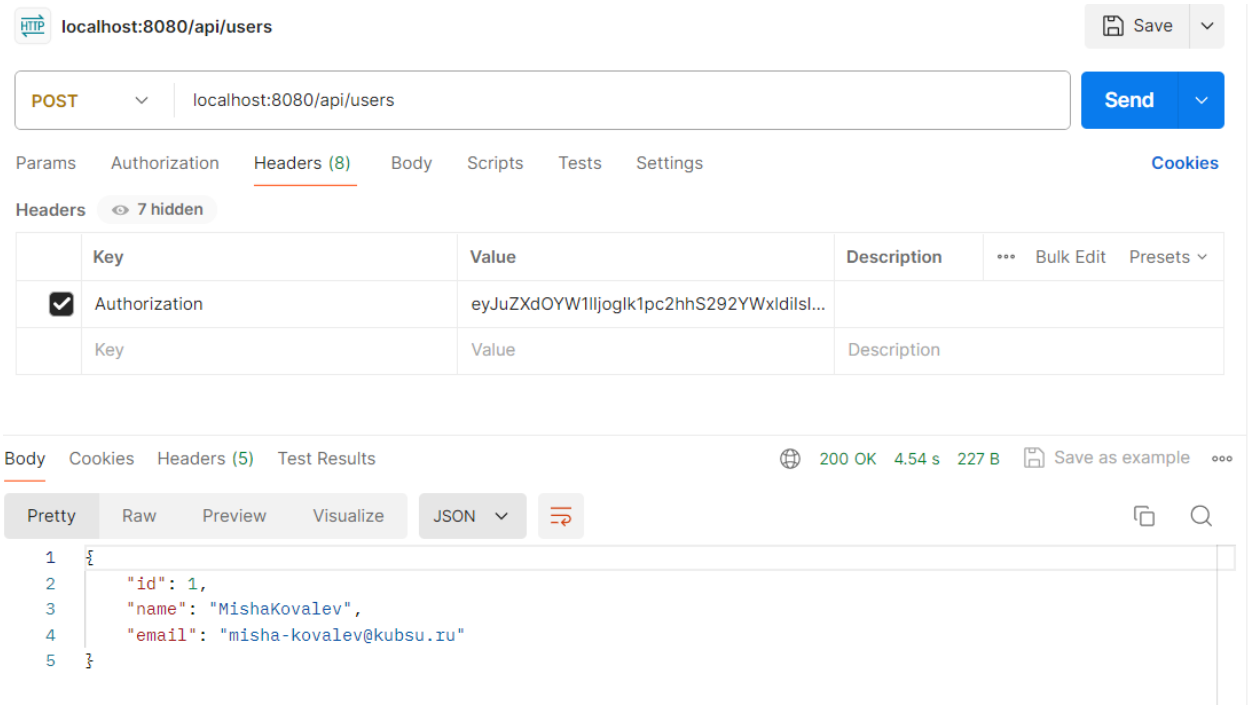


Рисунок 11 – Пример создания пользователя

3.3.4.2 GamesController

Контроллер для работы с играми предоставляет методы для создания и получения данных о них. Для функционирования методы GamesController используют GamesRepository, UserRepository и GamesService , которые были описаны выше.

Методы:

– createGame(String authorizationHeader, Long winId, Long loseId):

Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха, по полученным идентификаторам победителя и проигравшего, создает информация об игре. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getAllGames(String authorizationHeader): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха выводит информацию о всех существующих играх. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getGamesInfoById(String authorizationHeader, Long id): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха, по переданному идентификатору, возвращает информацию о статистике игр указанного пользователя. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getGamesInfo(String authorizationHeader): Выполняет те же функции, что и getGamesInfoById, за исключением того, что статистика игр пользователя возвращается только по переданным зашифрованным данным из заголовка авторизации.

В приложении Б представлена диаграмма последовательностей для каждого из методов GamesController. Данная диаграмма отражает передачу данных между разными компонентами приложения.

На рисунке 12 представлен результат выполнения http запроса на просмотр статистики игр указанного пользователя.

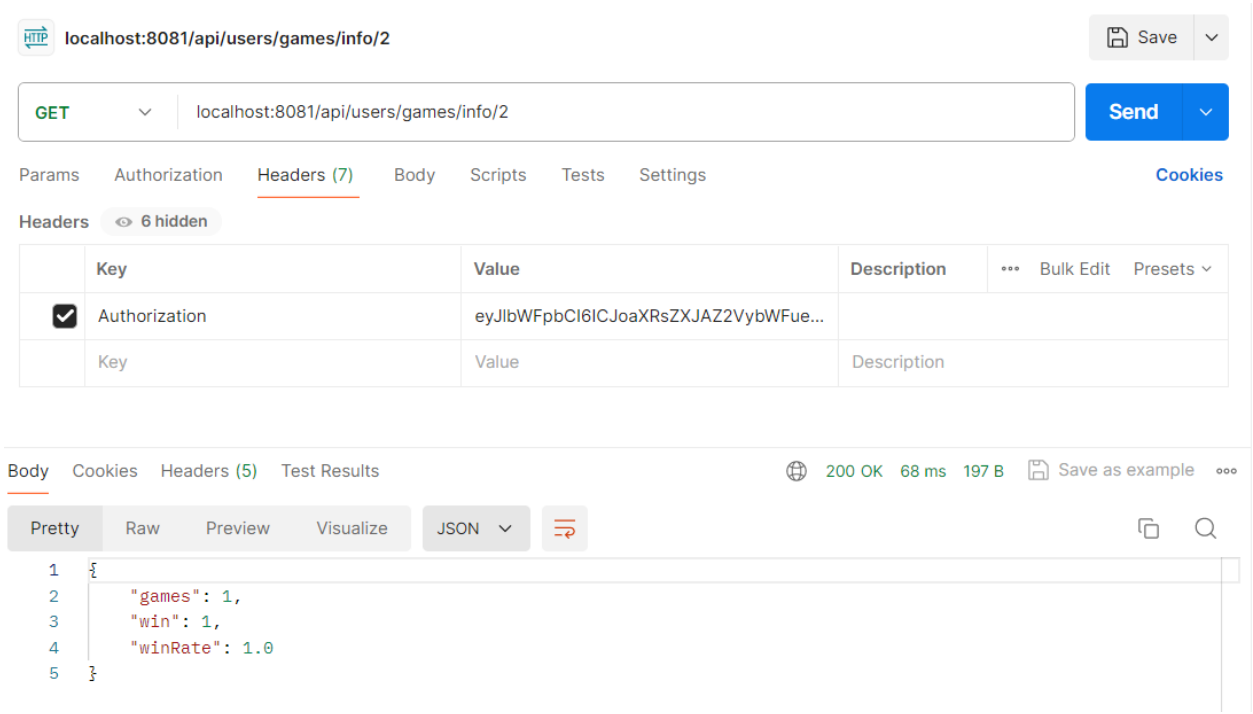


Рисунок 12 – Пример вывода статистики игр пользователя по указанному идентификатору

3.3.4.3 FriendsController

Контроллер для работы с друзьями пользователя предоставляет методы для создания, удаления и просмотра друзей. Для функционирования методы GamesController используют FriendsRepository, UserRepository и FriendsService, которые были описаны выше.

Методы:

– createFriend(String authorizationHeader, Long id): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха создает дружескую связь между текущим пользователем и пользователем с заданным идентификатором. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getSeveralFriends(String authorizationHeader, Long count): Получает список заданного количества друзей текущего пользователя. Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха выводит список заданного количества друзей

текущего пользователя. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getAllFriends(String authorizationHeader): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха выводит список всех друзей текущего пользователя. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– getFriendByName(String authorizationHeader, String nick): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха выводит список друзей текущего пользователя по заданному началу имени. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

– deleteFriend(String authorizationHeader, Long id): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха удаляет дружескую связь между текущим пользователем и пользователем с заданным идентификатором. В противном случае возвращает http – статус-код 401(Пользователь не авторизован).

В приложении В представлена диаграмма последовательностей для каждого из методов FriendsController. Данная диаграмма отражает передачу данных между разными компонентами приложения.

На рисунке 13 представлен результат выполнения http запроса на просмотр списка друзей текущего пользователя по заданному началу имени.

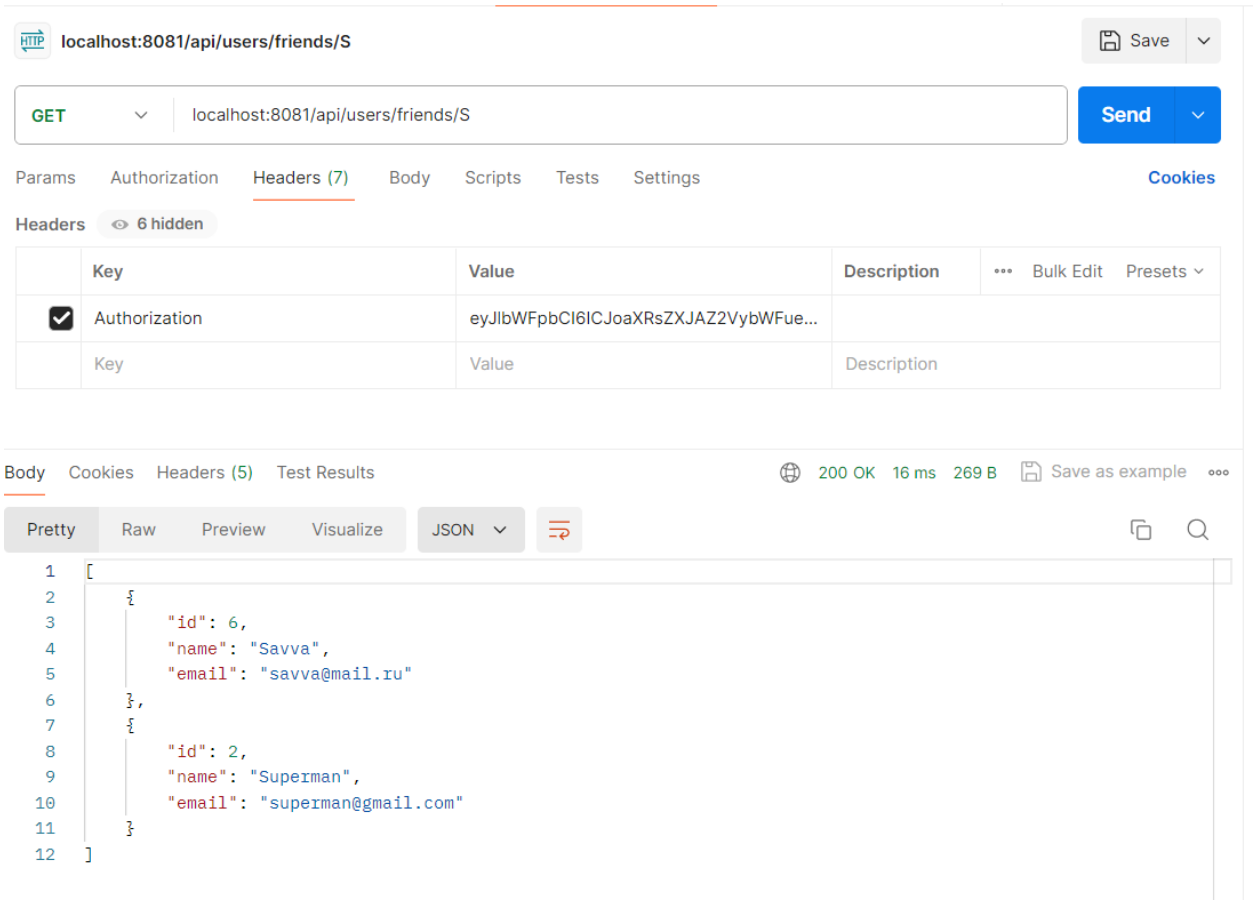


Рисунок 13 – Пример вывода списка друзей по заданному началу имени

3.3.4.4 VideoController

Контроллер предоставляет метод для отправки уникального видео. Для функционирования метода VideoController используются UserRepository и VideoService, которые были описаны выше.

Метод:

– mergeVideo(String authorizationHeader, Long count): Проверяет доступ пользователя по переданным зашифрованным данным из заголовка авторизации. В случае успеха генерирует и выводит видео. В противном случае возвращает http – статус-код 401(Пользователь не авторизован)(рис. 14 и 15).

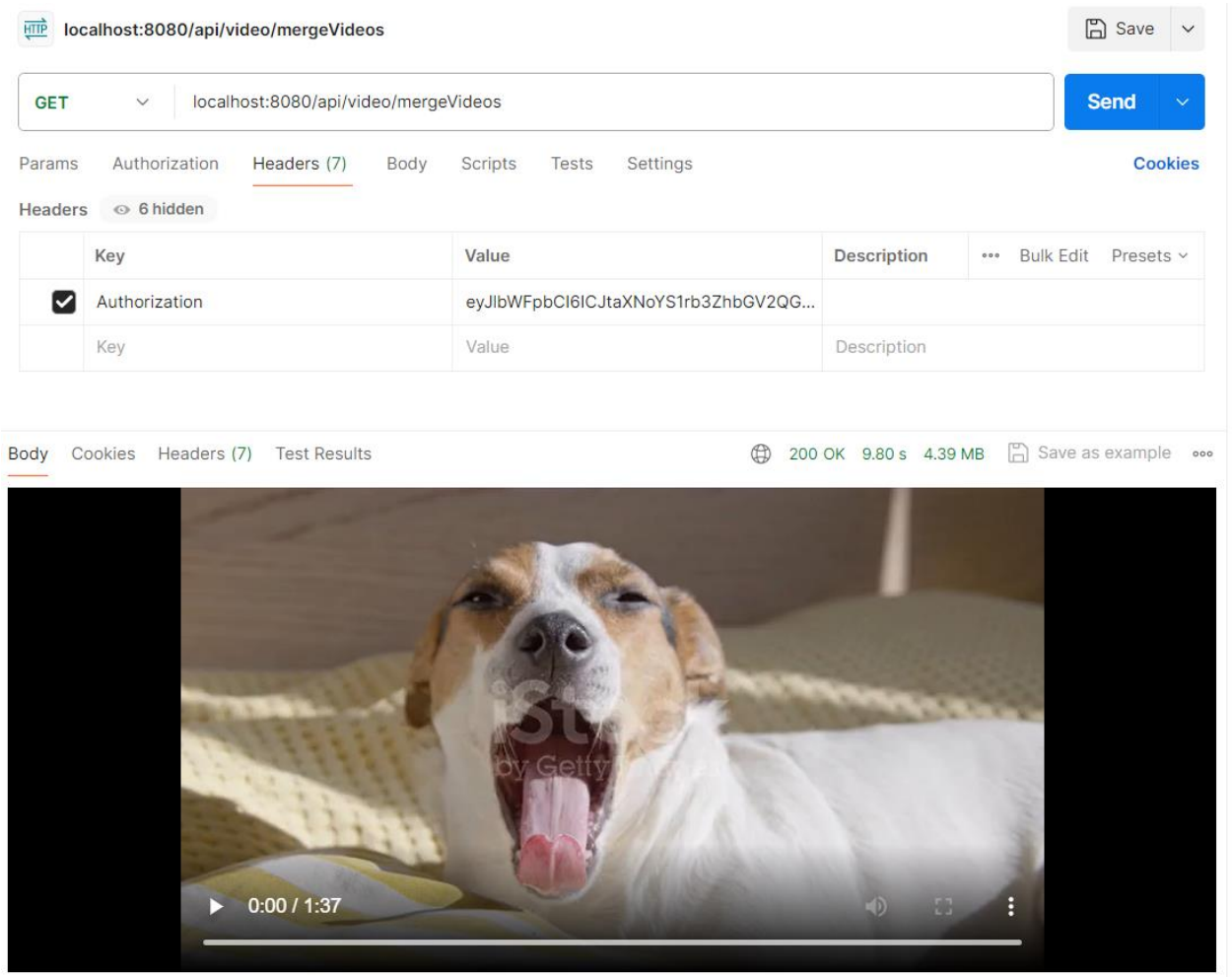


Рисунок 14 – Пример запроса видео

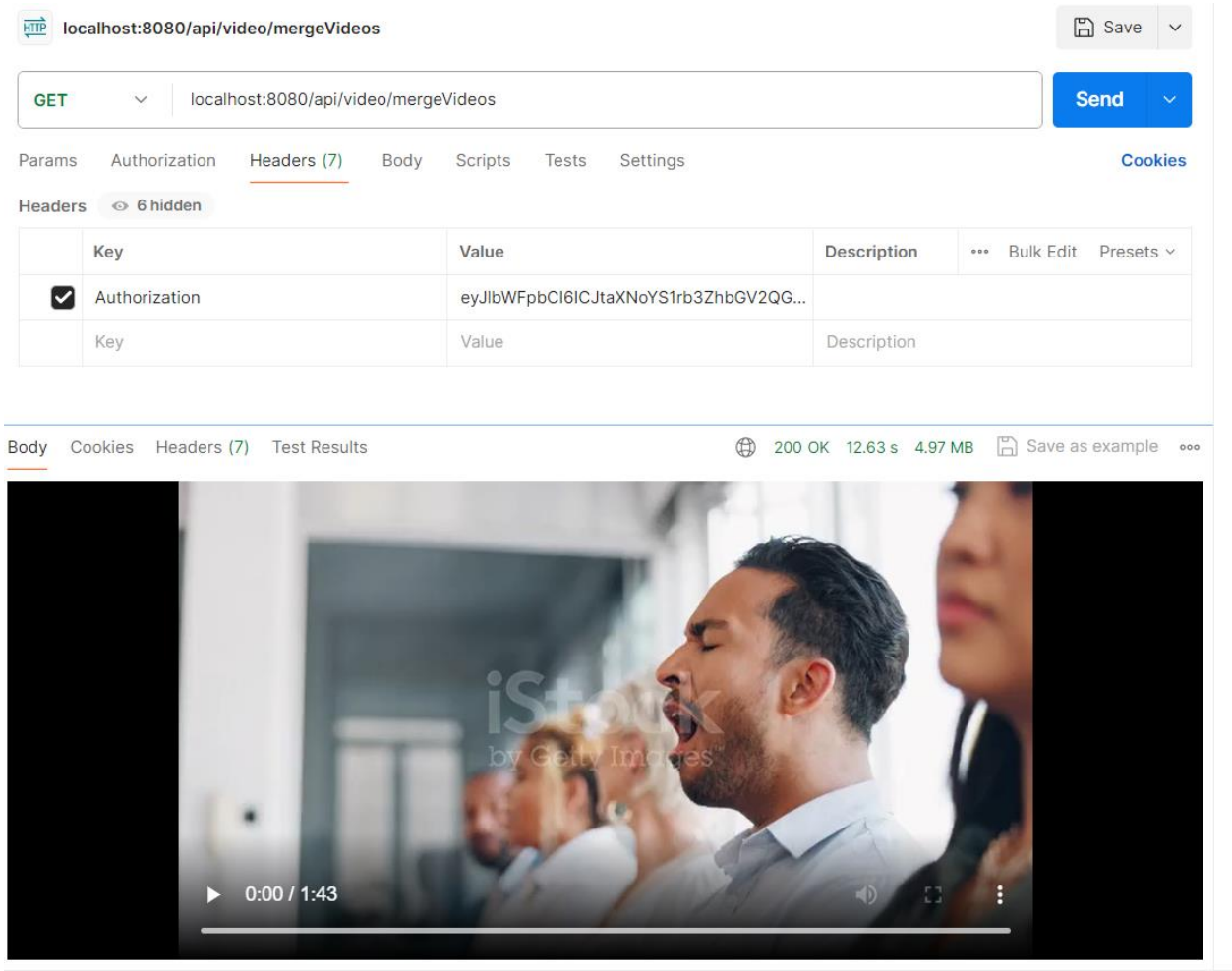


Рисунок 15 – Пример запроса видео

На рисунке 16 представлена диаграмма последовательностей для каждого из методов VideoController. Данная диаграмма отражает передачу данных между разными компонентами приложения.

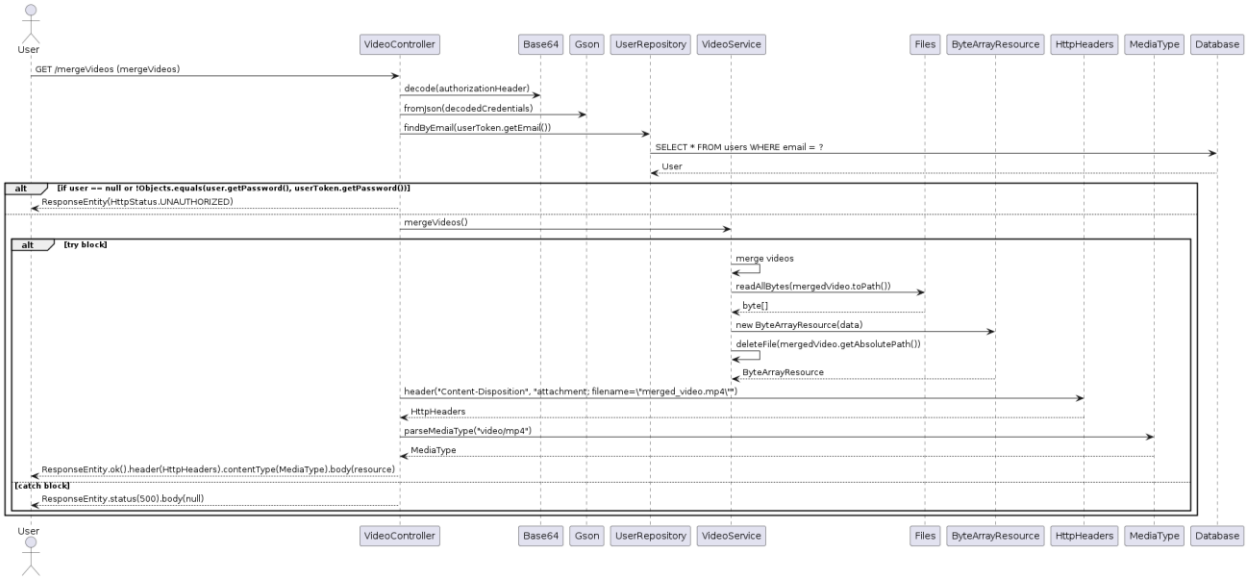


Рисунок 16 – Диаграмма последовательностей VideoController

3.3.4.5 Расшифровка пользовательских данных

Для каждого всех методов каждого контроллера данные пользователя передаются в зашифрованном base64 формате.

Примеры:

1) Передача пользовательских данных для авторизации и идентификации:

Зашифрованная строка, хранящаяся в заголовке авторизации:

eyJlbWFpbCI6ICJtaXNoYS1rb3ZhbGV2QGt1YnN1LnJ1IiwgInBhc3N3b3JkIjogIk1pc2hhMTIzIn0=

Объект с расшифрованными данными:

data = {

"email": "misha-kovalev@kubsu.ru",

"password": "Misha123"

}

2) Передача пользовательских данных для создания пользователя:

Зашифрованная строка, хранящаяся в заголовке авторизации:

eyJuZXdOYW1lIjogIk1pc2hhS292YWxldiIsICJuZXdFbWFpbCI6ICJtaXNoYS1rb3ZhbGV2QGt1YnN1LnJ1IiwgIm5ld1Bhc3N3b3JkIjogIk1pc2hhMTIzIn0=

Объект с расшифрованными данными:

data = {

"newName": "MishaKovalev",

"newEmail": "misha-kovalev@kubsu.ru",

"newPassword": "Misha123"

}

3) Передача пользовательских данных для изменения пользователя:

Зашифрованная строка, хранящаяся в заголовке авторизации:

eyJlbWFpbCI6ICJtaXNoYS1rb3ZhbGV2QGt1YnN1LnJ1IiwgInBhc3N3b3JkIjogIk1pc2hhMTIzIiwgIm5ld05hbWUiOiAiTWlzaGFLb3ZhbGV2VG9VcGRhdGUiLCAibmV3RW1haWwiOiAibWlzaGEtdXBkYXRlLWtvdmFsZXZAa3Vic3UucnUiLCAibmV3UGFzc3dvcmQiOiAiTWlzaGFVcGRhdGVkUGFzc3dvcmQifQ==

Объект с расшифрованными данными:

data = {

"email": "misha-kovalev@kubsu.ru",

"password": "Misha123",

"newName": "MishaKovalevToUpdate",

"newEmail": "misha-update-kovalev@kubsu.ru",

"newPassword": "MishaUpdatedPassword"

}

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе была реализована задача написания бэкенд части приложения с использованием нейронных сетей.

Была осуществлена разработка серверной части приложения «Зевок», а также создана нейронная сеть Yawn, определяющая зевания человека.

В итоге разработки удалось создать уникальное игровое приложения для борьбы с зеванием.

Стоит подчеркнуть актуальность выбранной темы. Игровое приложение борьбы с зевотой актуально в современных условиях по нескольким причинам. С ростом использования цифровых устройств и увеличением времени, проведенного за экранами, проблема усталости и скуки становится всё более заметной. Такие приложения могут не только развлекать пользователей, но и способствовать улучшению их общего состояния и производительности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое нейронная сеть?: сайт – URL: https://aws.amazon.com/ru/what-is/neural-network/ (дата обращения 15.10.2023).

2. Конспект лекции 1. Исторический обзор применения искусственного интеллекта. Обзор современных приложений искусственного интеллекта.: сайт – URL: http://academy21.ru/sveden/files/MR_B1.V.07_Prikladnye_sistemy_iskusstvennogo_intellekta.pdf (дата обращения 15.10.2023).

3. Полно связная нейронная сеть.: сайт – URL: https://proproprogs.ru/neural_network/struktura-i-princip-raboty-polnosvyaznyh-neyronnyh-setey (дата обращения 03.12.2023).

4. Рекуррентные нейронные сети.: сайт – URL: https://studfile.net/preview/7377680/page:8/ (дата обращения 03.12.2023).

5. Компьютерное зрение - что это и где применяется?: сайт – URL: https://neuro-core.ru/blogs/computer-vision-about (дата обращения 03.12.2023).

6. Памятка по проектированию систем: API стилей — REST, GraphQL, WebSocket, Webhook, RPC/gRPC, SOAP.: сайт – URL: https://hackernoon.com/ru/%D1%88%D0%BF%D0%B0%D1%80%D0%B3%D0%B0%D0%BB%D0%BA%D0%B0-%D0%BF%D0%BE-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD%D1%83-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B-API-%D1%81%D1%82%D0%B8%D0%BB%D0%B8-rest-%D0%B3%D1%80%D0%B0%D1%84%D0%BA%D0%BB-%D0%B2%D0%B5%D0%B1%D1%81%D0%BE%D0%BA%D0%B5%D1%82-%D0%B2%D0%B5%D0%B1%D1%85%D1%83%D0%BA-rpcgrpc-%D0%BC%D1%8B%D0%BB%D0%BE (дата обращения 05.02.2024).

7. Язык программирования Java: Его преимущества и особенности: сайт – URL: https://dzen.ru/a/ZQltAzEp7l2QP2Ls (дата обращения 15.02.2024).

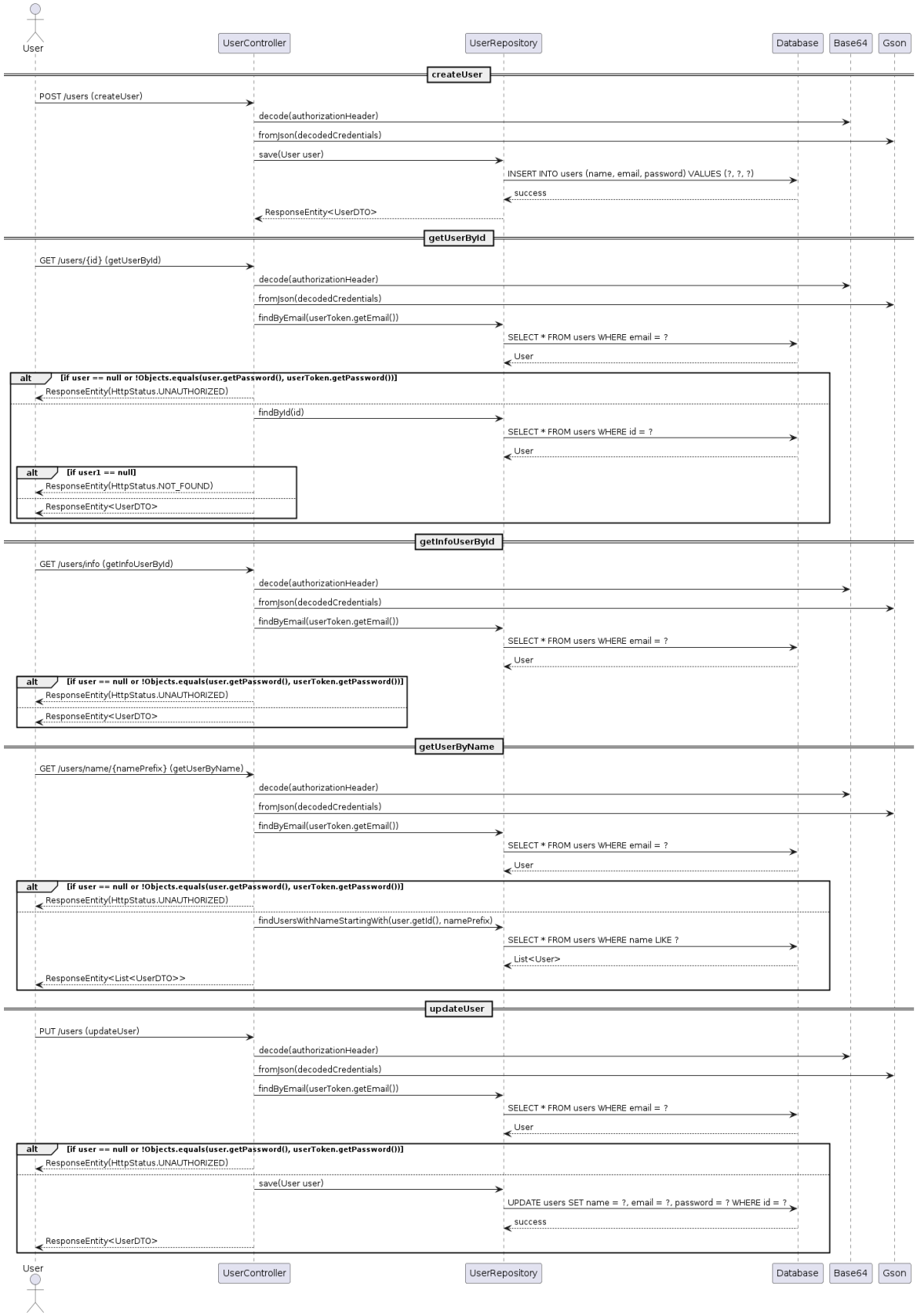
8. Плюсы и минусы использования Spring Boot: сайт – URL: https://scand.com/ru/company/blog/plyusy-i-minusy-ispolzovaniya-spring-boot/ (дата обращения 20.03.2024).

9. Что такое Docker?: сайт – URL: https://www.oracle.com/cis/cloud/cloud-native/container-registry/what-is-docker/ (дата обращения 23.03.2024).

10. Обнаружение эмоций: сайт – URL: https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer/data (дата обращения 04.11.2023)

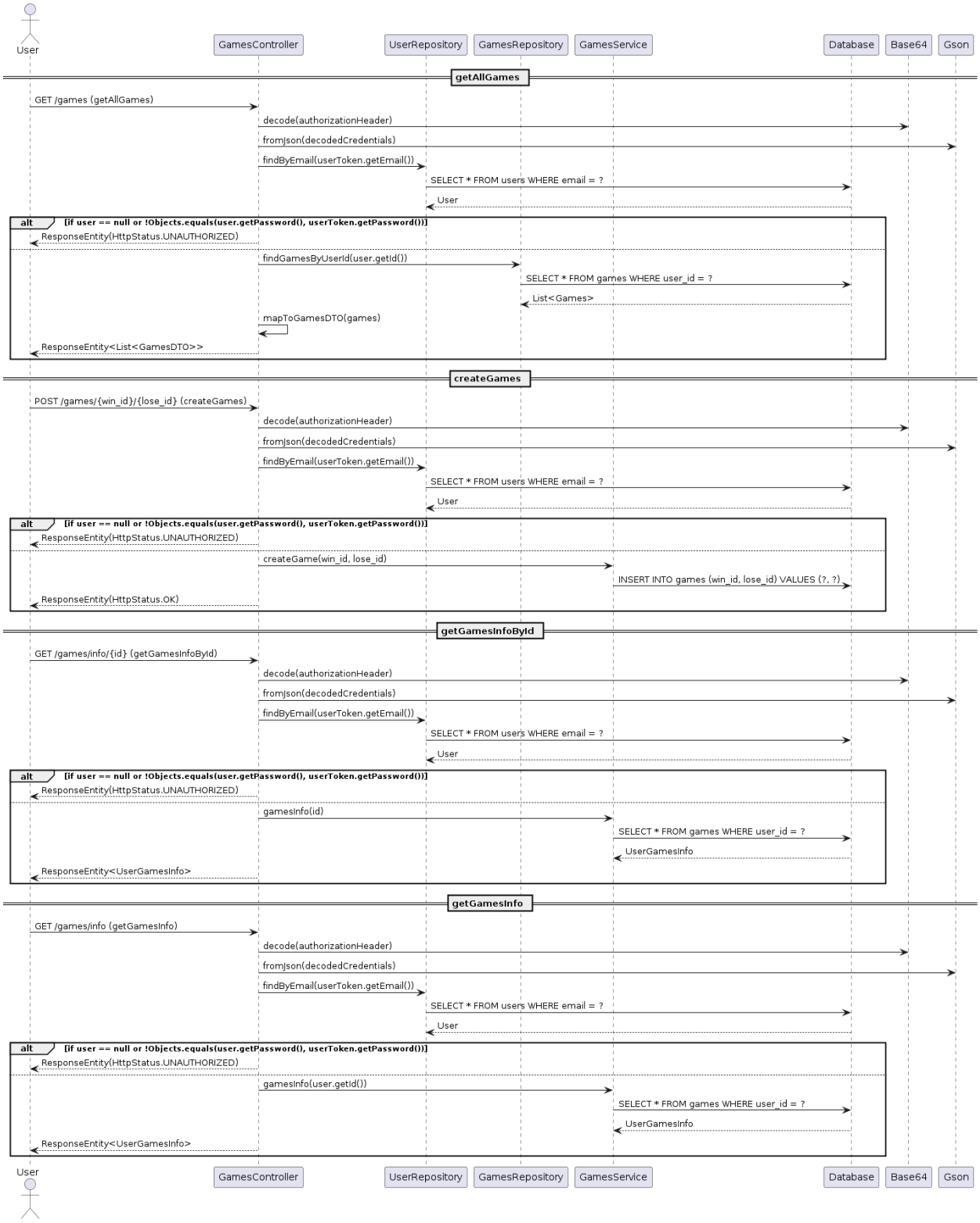
ПРИЛОЖЕНИЕ А

Диаграмма последовательностей UserController



ПРИЛОЖЕНИЕ Б

Диаграмма последовательностей GamesController



ПРИЛОЖЕНИЕ В

Диаграмма последовательностей FriendsController