БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Радиофизики и Компьютерных Технологий

Лабораторная работа №2

"Разработка технического задания на создание ИС"

«Интеллектуальная платформа для верификации мультимедийного контента в реальном времени»

Выполнили: студенты 4 курса, гр. 5ПИ
Шаковец Игнат Алексеевич,
Островский Иван Владимирович,
Сергеевич Владислав Дмитриевич

Проверил: Ломако Алексей Андреевич

г. Минск, 2025 г.

1. Общие положения

Проект «Разработка интеллектуальной платформы для верификации мультимедийного контента в реальном времени» направлен на создание комплексного программного решения, обеспечивающего автоматическую проверку подлинности медиа-контента (изображений, аудио, видео) с использованием методов искусственного интеллекта и блокчейн-технологий. Система разрабатывается на основании технико-экономического обоснования проекта. Проект актуален ввиду резкого роста объёма фейковых новостей и deepfake-контента, что угрожает доверию к информации. Основная цель проекта – снижение доли недостоверного контента в информационном пространстве за счёт быстрой верификации, что соответствует цели занять 5% рынка верификации мультимедиа в странах СНГ за 3 года.

Отрасль: MediaTech и информационная безопасность. Проект ориентирован на глобальный рынок решений по борьбе с дезинформацией и deepfake-контентом. Целевая аудитория включает СМИ, социальные сети, государственные структуры и IT-компании. Финансирование разработки предполагается в рамках инвестпроекта (плановая самоокупаемость – около 2,5 лет).

2. Назначение и цели ИС

Система предназначена для верификации достоверности мультимедийного контента в режиме реального времени. Её назначение – автоматически анализировать загружаемые или транслируемые медиа-данные (фото, видео, аудио) и определять степень их подлинности с помощью обученных нейронных моделей. Основная цель – существенное сокращение объёма дезинформации, публикуемой через цифровые каналы.

Ключевые цели и задачи ИС включают:

- Обеспечить потоковую (real-time) проверку видеопотока и статических изображений с минимальной задержкой.
- Повысить доверие пользователей и организаций к публикуемому контенту за счёт достоверной проверки.
- Создать удобный API/SaaS-сервис, позволяющий клиентам интегрировать проверки в свои системы и бизнес-процессы.
- Достигнуть коммерческих бизнес-целей: к 3 году выйти на 5% доли рынка СНГ и обеспечить самоокупаемость проекта через ~2,5 года.
- Предоставить гибкую подписочную модель монетизации (B2B и B2G).

3. Требования к системе

3.1. Функциональные требования

- **Проверка контента:** Система должна обеспечивать автоматическую верификацию мультимедийных файлов (изображений, аудио- и видеозаписей) в режиме реального времени. Пользователь загружает контент через веб-интерфейс или API, после чего система выполняет анализ метаданных и содержимого с использованием моделей искусственного интеллекта. По результатам проверки пользователю возвращается оценка подлинности (в процентах) и классификация состояния: *«подлинный», «возможный фейк» или «подделка»*.
- **Потоковый анализ видео:** Система должна обеспечивать приём и анализ видеопотоков в реальном времени по протоколам **RTSP** и **HLS**. Анализ выполняется по кадрам с интервалом 0,5–1 с, включая выделение лиц, сравнение с эталонными биометрическими шаблонами и детекцию артефактов deepfake. Результаты промежуточного анализа агрегируются и отображаются в виде таймлайна вероятности подлинности. По завершении обработки формируется итоговый отчёт. Для повышения достоверности применяется гибридный подход: **AI-анализ контента + проверка неизменности данных через блокчейн-журнал**.
- **Блокчейн-логирование:** Для обеспечения доказуемости и неизменности результатов каждая операция верификации регистрируется в распределённом журнале на основе **Hyperledger Fabric** (при необходимости — возможна миграция на Ethereum private network). В блок записываются: хэш файла/потока, идентификатор пользователя, дата и время проверки, результат анализа и контрольная подпись. Это обеспечивает аудит и защиту от последующего изменения данных. Доступ к журналу ограничен ролями и реализуется через административный интерфейс.

- **Интерфейсы:** Предусмотреть веб-интерфейс (пользовательский портал и админ-панель) для загрузки контента и просмотра отчётов. Обеспечить REST API для интеграции проверок в сторонние системы (медиа-платформы, социальные сети).

- **Управление пользователями:** Реализовать управление учётными записями и правами доступа: роли (администратор, аналитик, заказчик) с разграничением функций (загрузка контента, просмотр отчётов, настройка системы).

- **Отчёты и уведомления:** Генерировать отчёты о результатах проверки (в виде PDF/HTML) и при необходимости рассылать уведомления ответственным лицам. Предусмотреть хранение истории проверок.

- **Администрирование и мониторинг:** Обеспечить интерфейсы для администратора системы: настройка параметров (например, порогов достоверности), мониторинг состояния сервера и логирование операций

3.2. Нефункциональные требования

- **Производительность:** Время отклика системы при обработке одного объекта контента не должно превышать 2 секунд (Latency ≤ 2 с), чтобы гарантировать работу в реальном времени.

- **Точность:** Классификатор фейка должен обеспечивать точность не ниже 92% при проверке аудио/видео контента (вероятность правильной детекции ≥ 0.92), что соответствует современным требованиям к deepfake-детекторам.

- **Доступность:** Система должна быть доступна не менее 99.5% времени, обеспечивая круглосуточную работу сервисов и автоматическое резервирование основных компонентов.

- **Масштабируемость:** Архитектура системы должна поддерживать горизонтальное масштабирование для одновременной обработки множества запросов (не менее сотен конкурентных проверок).

- **Надёжность и отказоустойчивость:** При сбоях любого компонента должна обеспечиваться автоматическая избыточность и восстановление (сохранение данных в БД и блокчейне, дублирование сервисов).

- **Безопасность:** Обеспечить защиту данных на всех уровнях: шифрование хранимой и передаваемой информации, надёжная аутентификация и авторизация пользователей, защита от SQL-инъекций, XSS и других уязвимостей. Особое внимание – конфиденциальности обрабатываемого контента.

- **Юзабилити и доступность интерфейса:** Веб-интерфейс должен быть интуитивно понятным, поддерживать адаптивность под разные устройства. Поддерживать браузеры последних версий.

- **Совместимость и интеграция:** Система должна работать в облачных средах (Linux, Docker, Kubernetes) и поддерживать протоколы интеграции через API (REST, JSON).

- **Документируемость:** Все алгоритмы и модули должны иметь документацию (комментарии в коде, схемы архитектуры). Система должна соответствовать стандартам проектирования и документирования по ЕСКД/ЕСПД.

- **Локализация:** Интерфейс системы реализуется на русском языке как основном, с возможностью дальнейшего добавления английской версии для международных пользователей и партнёров.

3.3. Бизнес-требования

- **Коммерческая эффективность:** Проект должен обеспечить окупаемость инвестиций к концу 3-го года после запуска. Ожидается выход на самоокупаемость в течение ~2,5 лет после запуска сервиса (SaaS-модель).

- **Доля рынка:** Планируется занять порядка 5% рынка СНГ верификации мультимедийного контента в течение первых 3 лет, что накладывает требования по качеству сервиса и конкурентоспособности.

- **Монетизация:** Используется подписочная модель B2B и B2G – корпоративные клиенты и государственные структуры приобретают доступ к платформе по подписке.

- **SaaS и API:** Система должна быть реализована как SaaS-сервис с возможностью API-интеграции в инфраструктуру заказчиков. Предусматривается гибкая тарификация и партнёрские интеграции (например, с социальными сетями и новостными агрегаторами).

- **Конкурентные преимущества:** ИС должна использовать гибридный подход ИИ+блокчейн для обеспечения уникальности решения, а также обеспечивать лучшие показатели по задержке и стоимости по сравнению с аналогами.

- **Соответствие нормативам:** Необходимо соблюдать все законодательные и отраслевые требования к СМИ и ИТ (законы о распространении информации, GDPR/ локальные законы о защите данных), а также стандарты информационной безопасности.

3.4. Требования к видам обеспечения

Настоящие требования носят ориентировочный характер и могут уточняться в ходе проектирования и внедрения системы. Конфигурация и масштаб вычислительных ресурсов зависят от числа одновременных пользователей, интенсивности потоков и требуемого уровня производительности.

3.4.1. Техническое обеспечение

Система должна быть развёрнута в распределённой инфраструктуре (облачной или локальной), обеспечивающей горизонтальное масштабирование по мере роста нагрузки.
Рекомендуемые ориентировочные параметры:

- **Frontend-серверы (веб-доступ и API):**
  - 2–4 виртуальные машины (vCPU ≥ 4, RAM ≥ 16 ГБ, SSD ≥ 200 ГБ);
  - балансировщик нагрузки (NGINX / Kubernetes Ingress Controller).
- **Сервер(а) анализа мультимедиа:**
  - GPU-кластеры с ускорителями (NVIDIA RTX A6000 / A100 или эквивалент);
  - CPU ≥ 16 ядер, RAM ≥ 128 ГБ, SSD ≥ 2 ТБ;
  - масштабируемая очередь задач (RabbitMQ, Kafka).
- **Сервер баз данных и блокчейн-журналов:**
  - Отдельный узел PostgreSQL (репликация + резервное копирование);
  - Узлы Hyperledger Fabric для записи хэшей и аудита (не менее 3 пиров).
- **Сеть и инфраструктура:**
  - Пропускная способность сети ≥ 1 Гбит/с между узлами;
  - Наличие VPN или защищённого канала связи;
  - Источники бесперебойного питания и система резервирования.

3.4.2. Программное обеспечение

- ОС: Linux (Ubuntu Server 22.04 LTS или CentOS 9 Stream).
- Контейнеризация: Docker, Kubernetes.
- AI-фреймворки: PyTorch, TensorFlow.
- Очереди: RabbitMQ / Kafka.
- СУБД: PostgreSQL.
- Блокчейн: Hyperledger Fabric (возможна миграция на Ethereum Private Network).
- Web/API: FastAPI, Nginx, React (для интерфейса).
- Мониторинг и DevOps: Prometheus, Grafana, GitLab CI/CD.

3.4.3. Информационное обеспечение

- Репозиторий эталонных шаблонов и верифицированных медиафайлов.

- Базы данных с журналами проверок и хэшами.
- Наборы обучающих и тестовых данных для нейронных моделей.
- Конфигурационные и справочные данные (форматы, классификаторы, словари).

3.4.4. Организационное обеспечение

- Регламенты для всех ролей пользователей: администраторы, аналитики, заказчики.
- Процедуры управления доступом, учёта и удаления данных.
- Политики безопасности и конфиденциальности.
- Ответственные за эксплуатацию, резервное копирование и обновления системы.

3.4.5. Методическое обеспечение

- Руководство пользователя и администратора.
- Инструкции по установке, обновлению и диагностике.
- Методики тестирования и оценки достоверности верификации контента.
- Документы по организации резервирования, аудита и мониторинга.

4. Состав и сроки работ

1. **Анализ требований и проектирование архитектуры** – 0–2 месяца. Провести уточнение функциональных требований, спроектировать общую архитектуру системы, определить технологический стек.

2. **Разработка MVP системы (0–6 мес)**. Создание минимально рабочей версии платформы: реализация загрузки контента, базового анализа с использованием нейросетей и записи результатов.

3. **Интеграция блокчейн-модуля** – 4–7 месяц. Разработка и подключение модуля блокчейн-логирования проверок.

4. **Разработка пользовательского и административного интерфейса** – 4–8 месяц. Реализация веб-портала для загрузки контента, просмотра отчётов и администрирования, а также настройка REST API для интеграции.

5. **Тестирование и пилотные внедрения (7–12 мес)**. Проведение модульных, интеграционных и приёмочных испытаний с выборкой реальных данных. Доработка системы по результатам тестов. Выполнение пилотных запусков с первыми клиентами.

6. **Внедрение и вывод продукта на рынок (13–18 мес)**. Развёртывание облачной инфраструктуры, подготовка окружения эксплуатации. Публикация сервиса как SaaS-приложения, маркетинговая поддержка первых коммерческих запусков.

7. **Масштабирование и поддержка (19–30 мес)**. Расширение ресурсной инфраструктуры под рост нагрузки, оптимизация алгоритмов. Обеспечение достижения 5% доли рынка и финансовой самоокупаемости, заключение контрактов с новыми клиентами.
(Приведённые сроки ориентировочные и могут уточняться по ходу проекта в зависимости от ресурсов.)

5. Порядок приемки

Приёмка системы осуществляется по итогам приёмочных испытаний, подтверждающих выполнение всех функциональных и нефункциональных требований. Критериями приёмки являются: корректное выполнение ключевых функций (верификация медиа-контента), достижение SLA-показателей (время отклика, точность, доступность) и отсутствие критических ошибок. Выполнение требований подтверждается отчётами о модульных, интеграционных и системных тестах.

Приёмку выполняет комиссия, в состав которой входят представители разработчиков проекта и преподаватель (заказчик). Окончательное решение оформляется подписанным актом приёмки системы.

6. Требования к документации

В процессе разработки информационной системы разработчик обязан подготовить комплект проектной и эксплуатационной документации, обеспечивающий возможность внедрения, сопровождения и использования системы.

Документация должна быть оформлена в соответствии с требованиями ГОСТ 34.602-2020 и стандартов ЕСПД.

В состав подлежащих разработке документов входят:

- Техническое задание — описание целей, функций и требований к системе;
- Проектная документация — архитектура, структура модулей, схемы взаимодействия компонентов, описание API и БД;
- Эксплуатационная документация — руководство пользователя и администратора, инструкции по установке и настройке системы;
- Программная документация — комментарии в коде, описание конфигурации, схемы развёртывания и CI/CD;
- Тестовая документация — план тестирования, сценарии, протоколы испытаний и отчёты о результатах;
- Отчёт о внедрении и приёмке — перечень выполненных работ, результаты приёмочных испытаний, акт сдачи-приёмки.

Все документы должны быть доступны в актуальной версии в репозитории проекта и переданы заказчику в электронном виде (PDF и DOCX).