**Согласовано**

Прикладной администратор

Калмыкова А.П. \_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

Системный аналитик

Майоров В.Ю. \_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

**Техническое задание**

на выполнение работы по проектированию

инструмента «Техрадар»

Листов 55

Москва 2024

Содержание

[1. Введение 3](#_heading=h.gjdgxs)

[1.1. Термины и сокращения 3](#_heading=h.30j0zll)

[1.2. Общие сведения 4](#_heading=h.3znysh7)

[1.3. Организация-заказчик 5](#_heading=h.2et92p0)

[1.4. Исполнитель 5](#_heading=h.tyjcwt)

[1.5. Краткие сведения о компании 5](#_heading=h.3dy6vkm)

[1.6. Основания для выполнения работы 5](#_heading=h.1t3h5sf)

[1.7. Плановые сроки начала и окончания работ по созданию сайта 5](#_heading=h.mygm54xgaw55)

[2. Назначение и цели создания технического радара 5](#_heading=h.ffjnb8a55cdp)

[2.1. Общее описание 5](#_heading=h.2s8eyo1)

[2.2. Назначение 7](#_heading=h.5mciok3dk01r)

[2.3. Функциональные Возможности 8](#_heading=h.pc925zr7jwym)

[2.4. Цели создания 10](#_heading=h.17dp8vu)

[2.5. Целевая аудитория 10](#_heading=h.3rdcrjn)

[3. Требования к программе и программному изделию 12](#_heading=h.lnxbz9)

[3.1. Основные функциональные требования 12](#_heading=h.1ksv4uv)

[3.2. Нефункциональные требования 14](#_heading=h.qdxh2xow5w59)

[3.3. Пользовательские требования 15](#_heading=h.njdymwu8mhdc)

[3.4. Бизнес-требования 16](#_heading=h.mx0uchu6f2zu)

[3.5. Требования к разграничению доступа 18](#_heading=h.fu4n5ndpaig)

[3.6. Требования к полям формы добавления технологии 19](#_heading=h.1t0h57cb9gvj)

[3.7. Общие требования к дизайну пользовательского интерфейса 19](#_heading=h.44sinio)

[3.8. Требования к техническим средствам разработчика backend 20](#_heading=h.2jxsxqh)

[3.9. Требования к техническим средствам системного администрирования 21](#_heading=h.z337ya)

[4. Требования к программной документации 22](#_heading=h.4i7ojhp)

[4.1. Основные требования 22](#_heading=h.2xcytpi)

[5. Стадии и этапы разработки 24](#_heading=h.1ci93xb)

[5.1. Этап 1. Разработка технического задания 24](#_heading=h.3whwml4)

[5.3. Этап 2. Начальный этап проектирования 24](#_heading=h.2bn6wsx)

[5.4. Этап 3. Завершение технического проекта 25](#_heading=h.qsh70q)

[5.5. Этап 4. Завершение рабочего проекта 25](#_heading=h.3as4poj)

[5.6. Этап 5. Испытание программного обеспечения 25](#_heading=h.1pxezwc)

[5.7. Этап 6. Внедрение и сопровождение 26](#_heading=h.49x2ik5)

[6. Спецификация API 27](#_heading=h.t9ra883iyxd1)

[6.1. Описание API 46](#_heading=h.4x70ysht4wi2)

[6.2. Описание внешних интеграций 49](#_heading=h.uufbav79ytpn)

[6.3. Маппинг полей на UI к полям АРІ 49](#_heading=h.ehu8w5sqiuhy)

[6.4. Возможные виды ошибок, их коды и сообщения 52](#_heading=h.w2l55d109310)

[6.5. Описание свойств UI полей: редактируемость, видимость, обязательность 54](#_heading=h.kcud99jil8g0)

# 1. Введение

# 1.1. Термины и сокращения

Специальные термины, используемые в настоящем ТЗ, приведены в таблице ниже. Прочая техническая терминология понимается в соответствии с действующими стандартами и рекомендациями международных органов, ответственных за вопросы стандартизации в сети Интернет.

| Термин | Описание |
| --- | --- |
| Веб-интерфейс | Совокупность экранов и элементов управления системы, позволяющих пользователю, осуществляющему доступ к системе через веб-браузер, осуществлять поддержку и управление системой. |
| Технический радар (Техрадар) | Это список технологий, инструментов, платформ, языков и фреймворков, которые ранжированы по применимости в продукте |
| Фреймворк | Это набор инструментов, ускоряющих разработку приложений |
| ТЗ | Техническое задание |
| API | API (англ. application programming interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. |
| CSV | CSV (Comma-Separated Values) — это текстовый формат для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит поля, разделенные запятыми. Тип файлов предназначен для передачи объемных текстовых данных между различными программами и сервисами. |
| JSON | JSON (JavaScript Object Notation) — стандартный текстовый формат для хранения и передачи структурированных данных. Он основан на синтаксисе объекта в JavaScript, но не привязан к нему. |
| ПМИ | Программа и методика испытаний |

# 

# 1.2. Общие сведения

* Настоящий документ является техническим заданием на разработку практического кейса «Техрадар», определяет требования и порядок разработки практического кейса холдинга «T1».
* Настоящее техническое задание может быть откорректировано в процессе выполнения работ в установленном порядке.

# 1.3. Организация-заказчик

Заказчик работы – IT-холдинг Т1

# 1.4. Исполнитель

Исполнитель работы – Команда №33 IT-лагеря Т1

# 1.5. Краткие сведения о компании

Холдинг Т1 — один из лидеров российского ИТ-рынка, партнер ключевых производителей и разработчиков в сфере ИТ. Т1 предоставляет полный спектр услуг для реализации высокотехнологичных проектов для ключевых отраслей экономики.

# 1.6. Основания для выполнения работы

Данная работа осуществляется в рамках темы летнего буткемпа холдинга Т1 - «Разработка программного обеспечения Техрадар»

# 1.7. Плановые сроки начала и окончания работ по созданию сайта

Начало работ: 14.08.2024

Окончание работ: 31.08.2024

# 2. Назначение и цели создания технического радара

# 2.1. Общее описание

**Технический радар** — это визуальный инструмент, используемый для оценки и отслеживания использования технологий в компании или проекте. Он помогает группам разработчиков, архитекторов и руководителей принимать обоснованные решения о внедрении новых технологий, оптимизации существующих решений и отказе от устаревших и неэффективных инструментов.

Техрадар обеспечивает быструю аналитику при выборе основного стека технологий или вспомогательных инструментов при проектировании и реализации ИТ решения.

**Кольца на Радаре:**

Радар состоит из четырех концентрических колец, каждый из которых отражает степень готовности или рекомендаций по использованию той или иной технологии. Кольца следуют от центра к краям, символизируя движение технологий по мере их освоения и принятия в организации:

1. **Adopt (Принятие)** — Внутренний круг. Технологии, которые уже широко используются и доказали свою эффективность. Рекомендуется их повсеместное применение без особых ограничений.
2. **Trial (Испытание)** — Второе кольцо. Технологии, которые проходят активное тестирование. Они считаются перспективными, и их использование рекомендуется в ограниченных случаях или в пилотных проектах для более глубокого изучения.
3. **Assess (Оценка)** — Третье кольцо. Технологии, которые находятся на стадии оценки. Они потенциально интересны, но требуют дальнейшего анализа и изучения. Решение о их применении еще не принято.
4. **Hold (Заморозка)** — Внешний круг. Технологии, использование которых не рекомендуется или должно быть прекращено. Они могут быть устаревшими, небезопасными или неэффективными.

**Секторы на Радаре:**

Радар разделен на несколько секторов, каждый из которых отвечает за отдельную категорию технологий. В представленном радаре используются следующие сектора:

1. **Techniques (Методики)** — Отражает подходы, методы и процессы, используемые в разработке и поддержке ПО.
2. **Tools (Инструменты)** — Включает программные средства и утилиты, которые поддерживают процессы разработки и эксплуатации.
3. **Platforms (Платформы)** — Включает инфраструктурные компоненты и платформы, на которых разрабатываются и развертываются приложения.
4. **Languages (Языки программирования)** — Включает различные языки программирования и фреймворки, используемые для разработки ПО.

**Визуализация на Радаре:**

Каждая технология на радаре представлена в виде фигуры, расположенной в соответствующем кольце и секторе. Положение фигуры отражает степень уверенности в технологии, а также ее перспективность.

**Фигуры:**

* **Треугольники** обозначают технологии, которые **переместились вверх или вниз** в классификации (например, с Assess в Trial).
* **Звездочки** указывают на **новые** технологии, которые недавно были добавлены на радар.
* **Цвет точек**: Зеленый, Желтый, Красный, Салатовый – указывает на текущее состояние технологии и ее соответствие критериям данного кольца.

Этот радар позволяет быстро оценить, какие технологии стоит активно внедрять, какие еще требуют оценки, и от каких стоит отказаться.

# 2.2. Назначение

**Технический радар** предназначен для систематизации и визуализации информации о технологиях, используемых в компании. Он помогает руководству и техническим командам принимать информированные решения о выборе технологий для разработки и эксплуатации программного обеспечения. Основная цель радара — предоставить актуальную и наглядную картину текущих технологических трендов и их применимости в контексте компании или проекта.

# 2.3. Функциональные Возможности

1. **Визуализация технологического ландшафта:**
   * Радар позволяет наглядно представить различные технологии, инструменты, платформы и методики, которые находятся на разных стадиях жизненного цикла в компании: от стадии оценки до полного внедрения.
   * Кольца радара (Adopt, Trial, Assess, Hold) помогают классифицировать технологии по степени их зрелости и готовности к использованию.
2. **Мониторинг состояния технологий**:
   * Радар регулярно обновляется, что позволяет отслеживать изменение статуса технологий: их переход из одной категории в другую, добавление новых технологий или вывод устаревших.
   * Визуализация перемещения технологий на радаре позволяет отслеживать их прогресс или регресс, что помогает в принятии решений о дальнейших инвестициях в них.
3. **Оценка и рекомендации**:
   * Радар предоставляет рекомендации по использованию или прекращению использования определенных технологий на основе анализа их зрелости, стабильности и применимости в конкретных условиях.
   * Сектора радара позволяют оценивать и сравнивать технологии внутри различных категорий (например, инструменты против языков программирования), что упрощает процесс выбора.
4. **Поддержка стратегических решений**:
   * На основе данных радара технические лидеры могут формировать стратегии развития IT-инфраструктуры, планировать внедрение новых технологий, а также управлять рисками, связанными с устареванием или неопределенностью тех или иных технологий.
   * Радар также помогает согласовать техническую стратегию с бизнес-целями, обеспечивая баланс между инновациями и стабильностью.
5. **Коммуникация и обучение**:
   * Радар служит инструментом для эффективной коммуникации внутри компании, обеспечивая единое понимание текущего состояния технологий среди различных команд и уровней управления.
   * Он может использоваться для обучения новых сотрудников и повышения осведомленности всей команды о технологических трендах и предпочтениях компании.
6. **Поддержка экспериментов и инноваций**:
   * Категории Trial и Assess на радаре стимулируют проведение экспериментов с новыми технологиями, что позволяет компании оставаться на переднем крае инноваций.
   * Радар может служить отправной точкой для пилотных проектов и тестирования новых решений в контролируемых условиях.
   * Ведение каталога технологий, продуктов, инструментов (регистрация, категоризация, редактирование, обновление, версионирование, архивирование, удаление и т.д.)
   * Опросы и статистика
   * Голосование и рейтинг

Технический радар, таким образом, является важным инструментом для управления технологическим развитием компании, обеспечивая структурированный подход к выбору и внедрению технологий.

# 2.4. Цели создания

1. **Управление технологическим стеком**: Обеспечить прозрачность в управлении используемыми технологиями, их статусами и этапами жизненного цикла (внедрение, тестирование, оценка, приостановление использования).
2. **Обеспечение консистентности**: Уменьшить фрагментацию технологий внутри команды или компании, способствуя стандартизации и унификации подходов к разработке и эксплуатации систем.
3. **Мониторинг и контроль**: Отслеживать актуальность и эффективность используемых технологий, их эволюцию, а также изменения в предпочтениях команды.
4. **Поддержка принятия решений**: Оказать поддержку менеджерам и разработчикам в выборе оптимальных технологий для новых проектов и улучшений.
5. **Обучение и развитие**: Помочь команде ориентироваться в новых технологиях и инструментах, способствуя постоянному профессиональному развитию и обучению.

# 2.5. Целевая аудитория

1. **Руководители IT-отделов и архитекторы:**

* Для стратегического планирования и принятия решений о внедрении и использовании технологий в масштабах компании.
* Для управления технологическими рисками и бюджетами на внедрение новых инструментов.

1. **Технические лидеры и разработчики:**

* Для ориентации в доступных технологиях и их текущем статусе (Adopt, Trial, Assess, Hold).
* Для выбора инструментов и технологий в повседневной работе над проектами.

1. **Системные администраторы и DevOps-инженеры:**

* Для планирования и управления инфраструктурой с учетом текущих и будущих потребностей компании.
* Для оптимизации процессов развертывания и поддержки систем.

1. **Менеджеры проектов и продуктов:**

* Для согласования и планирования проектов с учетом технологических ограничений и возможностей.
* Для оценки рисков и сроков реализации проектов в зависимости от используемых технологий.

**5. Аналитики и консультанты:**

* Для анализа тенденций и изменений на рынке технологий.
* Для предоставления рекомендаций по оптимизации и модернизации существующих решений в компании.

**6. Новые сотрудники и стажеры:**

* Для быстрого погружения в технологический стек компании и понимания текущих стандартов и предпочтений в разработке и управлении проектами.

# 3. Требования к программе и программному изделию

# 3.1. Основные функциональные требования

Технический радар должен обладать следующими функциональными характеристиками:

**Управление технологическими решениями**:

* **Создание технологического решения**: Пользователи должны иметь возможность создавать новые записи о технологических решениях, вводя такие данные, как название, описание, категория, статус и дата внедрения.
* **Обновление информации**: Возможность редактирования существующих записей для актуализации данных о технологических решениях.
* **Удаление технологического решения**: Пользователи должны иметь возможность удалять записи о технологических решениях, которые больше не актуальны.
* **Просмотр всех технологических решений**: Система должна предоставлять возможность просмотра списка всех зарегистрированных технологических решений с возможностью фильтрации и сортировки.

**Оценка и анализ технологических решений**:

* **Создание и обновление оценок**: Пользователи должны иметь возможность добавлять или обновлять оценки технологических решений на основе определенных критериев.
* **Просмотр и фильтрация оценок**: Система должна предоставлять возможность просматривать оценки по различным категориям и фильтровать их по различным критериям (например, по категории, дате создания, рейтингу).
* **Анализ распределения оценок**: Встроенные инструменты для анализа и визуализации распределения оценок по категориям и отдельным технологическим решениям.

**Каталогизация технологий**:

* Возможность ведения и управления каталогом технологий, инструментов, платформ, языков программирования и фреймворков.
* Разделение технологий по категориям (например, Platforms, Techniques, Tools, Languages).

**Классификация технологий по статусу**:

Возможность классификации технологий по статусу:

* Adopt (Внедрить): Технологии, которые активно используются и рекомендованы к применению.
* Trial (Тестировать): Технологии, находящиеся на этапе тестирования и пилотного внедрения.
* Assess (Оценивать): Технологии, которые нужно оценивать для возможного будущего использования.
* Hold (Приостановить): Технологии, которые рекомендуется приостановить в использовании или отказаться от них.

Поддержка автоматической смены статуса технологии в зависимости от оценки и использования.

**Визуализация данных**:

* Графическое отображение технологий на радаре в виде круговой диаграммы с разделением на секции и кольца.
* Возможность отображения динамики изменения статуса технологий с помощью индикаторов (например, стрелки вверх или вниз).

**Опросы и статистика**:

* Возможность проведения опросов среди сотрудников для оценки использования и полезности технологий.
* Сбор и анализ статистических данных по использованию технологий в разных командах и проектах.

**Голосование и рейтинг**:

* Функционал голосования за технологии для определения их популярности и полезности среди пользователей.
* Система рейтингов для оценки эффективности технологий на основе мнений и опыта пользователей.

**Управление пользователями и правами доступа**:

* Возможность управления доступом к радару, настройка прав пользователей на просмотр и редактирование данных.
* Разделение прав доступа для различных ролей (например, администраторы, разработчики, менеджеры и т.д.).

**Обновление и поддержка актуальности данных**:

* Обновление информации о технологиях, базах данных и инструментах.
* Уведомления о необходимости пересмотра статуса технологии (например, через определенные промежутки времени).

# 3.2. Нефункциональные требования

Нефункциональные требования описывают характеристики системы, которые не связаны непосредственно с функциональностью, но важны для её успешного внедрения и эксплуатации.

1. **Производительность**:
   * **Время отклика**: Время отклика на любые пользовательские действия (например, создание или обновление записи) не должно превышать 3 секунд при средней нагрузке на систему.
   * **Поддержка большого объема данных**: Система должна быть способна эффективно обрабатывать и хранить данные о тысячах технологических решений без существенного падения производительности.
2. **Масштабируемость**:
   * Система должна поддерживать горизонтальное масштабирование для увеличения пропускной способности при росте числа пользователей или объема данных.
3. **Интерфейс и удобство использования**:
   * **Удобство интерфейса**: Пользовательский интерфейс должен быть интуитивно понятным и поддерживать стандарты UX/UI для корпоративных систем.
   * **Поддержка нескольких языков**: Интерфейс системы должен поддерживать несколько языков для пользователей из разных регионов.

# 3.3. Пользовательские требования

Пользовательские требования описывают ожидания и потребности различных групп пользователей системы.

1. **Топ-менеджмент**:
   * **Простота получения аналитической информации**: Доступ к сводным отчетам и статистике, а также возможность быстро получить полную информацию о текущем состоянии технологических решений.
   * **Доступ к данным в режиме реального времени**: Возможность просматривать актуальные данные и отчеты без задержек.
2. **Технические специалисты**:
   * **Управление технологическими решениями**: Интуитивно понятный интерфейс для создания, обновления и удаления записей о технологических решениях.
   * **Доступ к оценкам и статистике**: Возможность просматривать и анализировать оценки, распределение решений по категориям и статистику их использования.
3. **Аналитики**:
   * **Доступ к инструментам анализа**: Возможность глубоко анализировать данные о технологических решениях, сравнивать их и выявлять тренды.
   * **Гибкость в генерации отчетов**: Возможность настраивать параметры отчетов и выводить данные в удобных форматах.
4. **ИТ-администраторы**:
   * **Управление пользователями и доступом**: Возможность управлять ролями пользователей, настраивать уровни доступа и контролировать безопасность системы.
   * **Мониторинг и отчетность по безопасности**: Доступ к журналам активности и инструментам для мониторинга безопасности системы.

# 3.4. Бизнес-требования

Бизнес-требования описывают цели, которые должна достигнуть система "Техрадар" с точки зрения бизнеса. Эти требования отражают, как система должна способствовать достижению стратегических целей компании, оптимизировать процессы и приносить дополнительную ценность.

1. Оптимизация процесса принятия решений

* Цель: Обеспечить топ-менеджмент и ключевых лиц, принимающих решения, актуальной и достоверной информацией о технологических решениях, чтобы они могли принимать обоснованные и стратегически важные решения.
* Задачи:
  + Сокращение времени на анализ информации о технологических решениях.
  + Уменьшение рисков за счет своевременного выявления неэффективных решений и их замены.

2. Повышение эффективности управления технологическим портфелем

* Цель: Создание единого источника данных о всех технологических решениях, внедренных в компании, с целью упрощения управления и мониторинга их состояния.
* Задачи:
  + Централизация данных о технологических решениях, что позволит избежать дублирования информации и ошибок.
  + Автоматизация процессов обновления и анализа данных о решениях.

3. Повышение конкурентоспособности компании

* Цель: Использование "Техрадара" для обеспечения конкурентных преимуществ за счет быстрой адаптации к новым технологиям и своевременного внедрения инновационных решений.
* Задачи:
  + Мониторинг новых технологических решений и трендов, чтобы оперативно внедрять их в бизнес-процессы.
  + Обеспечение постоянного обновления технологического портфеля компании в соответствии с последними достижениями в отрасли.

4. Обеспечение соответствия нормативным требованиям и стандартам

* Цель: Гарантировать, что все технологические решения, используемые в компании, соответствуют нормативным требованиям и стандартам, что минимизирует риски и возможные штрафы.
* Задачи:
  + Встроенный контроль соответствия решений законодательным требованиям.
  + Автоматизация отчетности и обеспечение прозрачности для внутренних и внешних аудитов.

5. Улучшение взаимодействия между подразделениями

* Цель: Повысить уровень сотрудничества между различными подразделениями компании за счет создания общей платформы для обмена информацией о технологических решениях.
* Задачи:
  + Обеспечение доступа к единой базе данных для всех заинтересованных сторон.
  + Создание механизмов для обмена опытом и лучшими практиками между подразделениями.

# 3.5. Требования к разграничению доступа

Информация, размещаемая на сайте, является общедоступной.

Пользователей сайта можно разделить на 3 части в соответствии с правами доступа:

1. Посетители

2. Пользователи (сотрудник Заказчика)

3. Администратор (сотрудник Исполнителя/Заказчика)

Посетители имеют доступ только к общедоступной части сайта. Доступ к административной части имеют пользователи с правами администратора. Администратор может только добавлять технологии. Пользователи имеют право голосовать за ту или иную технологию по десятибалльной шкале. Доступ к пользовательским и административным возможностям должен осуществляться с использованием уникального логина и пароля.

Для обеспечения защиты от несанкционированного доступа к административной части при составлении паролей рекомендуется придерживаться следующих правил:

1. Длина пароля должна быть не менее 5 и не более 50 символов.

2. Пароль должен состоять из цифр (от 0 до 9) и латинских букв (от a-z и A-Z) в разных регистрах; желательно включать в пароль специальные символы, имеющиеся на клавиатуре (разрешены символ пробела, точки и нижнего подчеркивания). Кириллица не разрешена.

3. Для логина аналогичные правила.

# 3.6. Требования к полям формы добавления технологии

1. Длина поля “Название технологии” должна быть не менее 3 и не более 50 символов.

2. Разрешены символы: цифры (от 0 до 9), латинские буквы (от a-z и A-Z) в разных регистрах; символ пробела, точки и нижнего подчеркивания, запятая, плюс, минус). Кириллица не разрешена.

3. Производить проверку поля “Ссылка” на существование и корректность ссылки на технологию.

4. Длина поля “Ссылка” должна быть не менее 3 и не более 50 символов.

5. Разрешены символы: цифры (от 0 до 9), латинские буквы (от a-z и A-Z), кириллица (а-я, А-Я) в разных регистрах; специальные символы). Кириллица не разрешена.

# 3.7. Общие требования к дизайну пользовательского интерфейса

Дизайн пользовательского интерфейса должен адаптироваться под основные разрешения экранов настольных (десктоп), портативных (ноутбук) и планшетных устройств. Максимальное разрешение по горизонтали - 1920px, минимальное - 1024px. Интерфейс системы, включая графики и диаграммы, должен адаптироваться под разрешение экрана.

Элементы интерфейса (пункты меню, кнопки, поля ввода в формах, раскрывающиеся списки, и т.д.) должны адаптироваться по размеру под устройство, на котором просматривается система, и под основной сценарий использования данного устройства.

Дизайн пользовательского интерфейса должен быть разработан с учетом принципа сохранения работоспособности при потере части функциональности системы. Некоторые элементы дизайна могут иметь упрощенный вид, но основной функционал должен оставаться доступным.

Дизайн пользовательского интерфейса должен корректно отображаться и работать на настольных компьютерах, ноутбуках, планшетных устройствах. Предусмотреть версию для слабовидящих, используя контрастные цвета, крупные шрифты. Также (по возможности) предусмотреть темную и светлую тему для frontend.

# 3.8. Требования к техническим средствам разработчика backend

Для эффективной работы разработчика backend необходимы следующие технические средства:

* **Производительное оборудование:**
  + Процессор с высокой тактовой частотой и многоядерностью для выполнения сложных вычислительных задач.
  + Оперативная память объемом не менее 16 ГБ, желательно 32 ГБ для работы с большими проектами и виртуализацией.
  + Быстрый SSD-накопитель для хранения исходного кода, библиотек и других ресурсов, минимальный объем — 512 ГБ.
* **Операционная система:**
  + Поддержка разработки на популярных платформах (Linux, macOS, Windows), с возможностью работы в виртуальных средах или контейнерах (Docker).
* **Сетевые ресурсы:**
  + Высокоскоростное подключение к сети для работы с удаленными репозиториями, облачными сервисами и инструментами CI/CD.
* **Инструменты для работы с кодом:**
  + Современные IDE (например, IntelliJ IDEA, Visual Studio Code) с поддержкой необходимых языков и фреймворков.
  + Средства для управления версиями кода (Git) и работы с репозиториями (GitHub, GitLab).
* **Средства для тестирования:**
  + Инструменты для написания и автоматизации тестов (JUnit, PyTest, Postman).
  + Локальные серверы для развертывания тестовой среды.
* **Средства для взаимодействия и документации:**
  + Приложения для командного взаимодействия и управления проектами (Jira, Confluence, Slack).
  + Средства для написания и хранения технической документации.

# 3.9. Требования к техническим средствам системного администрирования

Для системного администрирования необходимо следующее техническое обеспечение:

1. Рабочая станция:

* Высокопроизводительное оборудование, аналогичное требованиям для backend-разработчиков (многоядерный процессор, минимум 16 ГБ ОЗУ, SSD).
* Возможность работы с несколькими мониторами для мониторинга и управления системами в реальном времени.

2. Операционные системы и среды:

* Поддержка основных серверных операционных систем (Linux, Windows Server) и инструментов управления ими.
* CentOS 7.9 как базовая операционная система.
* Возможность работы с виртуализацией (Oracle VM VirtualBox) и контейнеризацией (Docker).

3. Средства управления сетевой инфраструктурой:

* Nginx как веб-сервер.

4. Инструменты для автоматизации задач:

* Скриптовые языки программирования (Python) для автоматизации задач и написания утилит.

5. Резервное копирование и восстановление:

* Инструменты для создания и управления резервными копиями данных и конфигураций (Veeam, Acronis).

# 4. Требования к программной документации

# 4.1. Основные требования

1. **Полнота и детализация:**
   * Документация должна содержать исчерпывающую информацию о всех ключевых аспектах программного обеспечения, включая архитектуру, функциональные характеристики, интерфейсы, используемые технологии и инструменты.
   * Необходимо описывать как высокоуровневые концепции, так и технические детали реализации, включая алгоритмы, структуры данных и логику взаимодействия компонентов.
2. **Актуальность:**
   * Документация должна регулярно обновляться в соответствии с изменениями в программном обеспечении. Важно поддерживать актуальность всех разделов, чтобы пользователи и разработчики могли полагаться на нее при работе с системой.
   * В документации необходимо четко указывать версию программного обеспечения, к которой она относится.
3. **Структурированность:**
   * Документация должна быть логически структурирована и разделена на тематические разделы, чтобы пользователи могли быстро находить нужную информацию.
   * Использование оглавления, индексирования и перекрестных ссылок для облегчения навигации по документу.
4. **Ясность и доступность:**
   * Язык документации должен быть понятным и доступным для целевой аудитории. Следует избегать излишней терминологии и использовать примеры для иллюстрации сложных концепций.
   * Документация должна быть доступна в удобном для чтения и использования формате, как в электронном, так и в печатном виде.
5. **Документация по установке и настройке:**
   * Включение детальных инструкций по установке, настройке и конфигурации программного обеспечения, а также требований к среде исполнения.
   * Описание возможных проблем и способов их решения при установке и настройке.
6. **Документация по использованию:**
   * Подробное описание функционала программного обеспечения, его интерфейсов, а также инструкций по использованию.
   * Включение разделов с примерами использования и типовыми сценариями.
7. **Документация по интеграции:**
   * Описание способов интеграции с внешними системами и сервисами, требования к API и форматам данных.
   * Инструкции по настройке и поддержке интеграций.
8. **Технические спецификации:**
   * Подробное описание архитектуры системы, используемых технологий, протоколов, библиотек и других технических аспектов.
   * Схемы архитектуры, диаграммы классов и последовательностей, а также другие визуальные представления структуры системы.
9. **История изменений (Changelog):**
   * Ведение истории изменений (Changelog), в которой фиксируются все обновления и изменения в программном обеспечении.
   * Указание версий, дат релиза и краткого описания внесенных изменений.
10. **Поддержка и сопровождение:**
    * Документация должна содержать информацию о поддержке программного обеспечения, включая контакты технической поддержки, описание типичных проблем и часто задаваемых вопросов (FAQ).
    * Рекомендации по регулярному обновлению документации и процедурам ее сопровождения.

# 5. Стадии и этапы разработки

# 5.1. Этап 1. Разработка технического задания

В техническом задании указываются основные характеристики будущего продукта, ожидаемые результаты, критерии приемки, сроки выполнения работ, и планируемые ресурсы. Этот документ служит основой для дальнейшего проектирования и разработки.

# 5.3. Этап 2. Начальный этап проектирования

На втором этапе начинается проектирование программного обеспечения. Основная цель этого этапа — разработка архитектуры системы, определение ее компонентов и их взаимодействий. Разработчики анализируют требования из технического задания и создают архитектурные решения, которые будут отвечать заявленным требованиям.

На этом этапе также проводится выбор технологий, инструментов, языков программирования и фреймворков, которые будут использоваться в проекте. В результате выполнения начального этапа проектирования формируется базовая архитектура системы и создается ее прототип.

# 5.4. Этап 3. Завершение технического проекта

На третьем этапе осуществляется завершение технического проекта, который включает в себя детальную проработку всех аспектов системы. Здесь проводится разработка детализированных технических схем, спецификаций и документации, которая описывает работу всех компонентов системы.

В это время также завершается проектирование пользовательского интерфейса, базы данных и интеграций с другими системами. Этот этап важен для обеспечения готовности проекта к непосредственной разработке и последующему внедрению.

# 5.5. Этап 4. Завершение рабочего проекта

Четвертый этап включает в себя завершение рабочего проекта, на котором происходит реализация всех разработанных решений и подготовка продукта к испытаниям.

Разработчики программного обеспечения заканчивают кодирование, проводят внутренние тестирования, устраняют выявленные ошибки и оптимизируют производительность. На этом этапе продукт должен быть полностью готов к проведению официальных испытаний, что включает в себя финальную сборку и подготовку всей необходимой документации.

# 5.6. Этап 5. Испытание программного обеспечения

На заключительном этапе проводится испытание программного обеспечения с целью проверки его соответствия всем требованиям технического задания и проектной документации. Испытания включают в себя функциональное, нагрузочное, интеграционное и другие виды тестирования.

Основная цель этого этапа — убедиться, что программное обеспечение работает стабильно, эффективно и безопасно в различных условиях эксплуатации. По результатам испытаний составляется отчет, на основе которого принимается решение о готовности продукта к внедрению и эксплуатации.

# 5.7. Этап 6. Внедрение и сопровождение

На этом этапе осуществляется внедрение разработанного программного обеспечения в рабочую среду компании. Включает в себя следующие задачи:

1. **Развертывание**: Настройка и установка программного обеспечения на целевых серверах или облачных платформах, интеграция с существующими системами.
2. **Обучение пользователей**: Проведение обучающих сессий для сотрудников, которым предстоит работать с новой системой, обеспечение их необходимой документацией и поддержкой.
3. **Начальная поддержка**: Предоставление оперативной технической поддержки на первых этапах использования продукта, решение возникающих проблем и внесение мелких исправлений.
4. **Мониторинг и обратная связь**: Сбор данных о производительности и надежности системы, а также отзывов от пользователей для дальнейшего улучшения продукта.
5. **Планирование дальнейших обновлений**: Разработка плана регулярных обновлений и улучшений на основе полученной информации, с целью поддержания актуальности и функциональности системы в долгосрочной перспективе.

# 6. Спецификация API

{

"openapi": "3.0.1",

"info": {

"title": "Case Management Service",

"description": "Claim Event Information",

"version": "1.0"

},

"servers": [

{

"url": "http://localhost:8080",

"description": "Generated server url"

}

],

"security": [

{

"bearerAuth": []

}

],

"paths": {

"/api/tech/updateTechSolution": {

"put": {

"tags": [

"tech-solution-controller"

],

"operationId": "updateTechSolution",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/TechSolution"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "string"

}

}

}

}

}

}

},

"/api/tech/updateEffectivenessOnClick": {

"post": {

"tags": [

"tech-solution-controller"

],

"operationId": "updateEffectivenessOnClick",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK"

}

}

}

},

"/api/tech/createSolution": {

"post": {

"tags": [

"tech-solution-controller"

],

"operationId": "createSolution",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/TechSolutionDTO"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "string"

}

}

}

}

}

}

},

"/api/score/addOrUpdate": {

"post": {

"tags": [

"score-controller"

],

"operationId": "addOrUpdateUserScore",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/AddScoreRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "object"

}

}

}

}

}

}

},

"/api/auth/token": {

"post": {

"tags": [

"auth-controller"

],

"operationId": "getNewAccessToken",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/JwtRefreshRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "object"

}

}

}

}

}

}

},

"/api/auth/register": {

"post": {

"tags": [

"auth-controller"

],

"operationId": "register",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/RegisterRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "string"

}

}

}

}

}

}

},

"/api/auth/refresh": {

"post": {

"tags": [

"auth-controller"

],

"operationId": "getNewRefreshToken",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/JwtRefreshRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "object"

}

}

}

}

}

}

},

"/api/auth/login": {

"post": {

"tags": [

"auth-controller"

],

"operationId": "login",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/LoginRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "object"

}

}

}

}

}

}

},

"/api/tech/getAllSolutions": {

"get": {

"tags": [

"tech-solution-controller"

],

"operationId": "getAllTechSolution",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"$ref": "#/components/schemas/TechSolutionOnRadarResponse"

}

}

}

}

}

}

},

"/api/statistic/year": {

"get": {

"tags": [

"statistic-controller"

],

"operationId": "getYearStatistic",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Characteristics"

}

}

}

}

}

}

}

},

"/api/statistic/score/distribution/{tech\_id}": {

"get": {

"tags": [

"statistic-controller"

],

"operationId": "getScoreDistribution",

"parameters": [

{

"name": "techId",

"in": "query",

"required": true,

"schema": {

"type": "integer",

"format": "int64"

}

}

],

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "integer",

"format": "int32"

}

}

}

}

}

}

}

},

"/api/statistic/month": {

"get": {

"tags": [

"statistic-controller"

],

"operationId": "getMonthStatistic",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Characteristics"

}

}

}

}

}

}

}

},

"/api/statistic/characteristics": {

"get": {

"tags": [

"statistic-controller"

],

"summary": "Write techSolution characteristics into DB manually",

"operationId": "snapshotCharacteristics",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK"

}

}

}

},

"/api/score/getScores": {

"get": {

"tags": [

"score-controller"

],

"operationId": "getScores",

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "array",

"items": {

"$ref": "#/components/schemas/ScoreDTO"

}

}

}

}

}

}

}

},

"/api/score/getScoresByTechSolutionsCategory": {

"get": {

"tags": [

"score-controller"

],

"operationId": "getScoresByTechSolutionsCategory",

"parameters": [

{

"name": "category",

"in": "query",

"required": true,

"schema": {

"type": "string",

"enum": [

"LANGUAGES",

"TOOLS",

"TECHNIQUES",

"PLATFORMS"

]

}

}

],

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"type": "array",

"items": {

"$ref": "#/components/schemas/ScoreDTO"

}

}

}

}

}

}

}

},

"/api/score/getNotScoredTechSolutionsByCategory/{category}": {

"get": {

"tags": [

"score-controller"

],

"operationId": "getTechSolutionsByCategory",

"parameters": [

{

"name": "category",

"in": "path",

"required": true,

"schema": {

"type": "string",

"enum": [

"LANGUAGES",

"TOOLS",

"TECHNIQUES",

"PLATFORMS"

]

}

}

],

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK",

"content": {

"\*/\*": {

"schema": {

"$ref": "#/components/schemas/TechSolutionOnRadarResponse"

}

}

}

}

}

}

},

"/api/tech/delete/{id}": {

"delete": {

"tags": [

"tech-solution-controller"

],

"operationId": "deleteTechSolution",

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "integer",

"format": "int64"

}

}

],

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK"

}

}

}

},

"/api/score/delete": {

"delete": {

"tags": [

"score-controller"

],

"operationId": "deleteUserScore",

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/DeleteScoreRequest"

}

}

},

"required": true

},

"responses": {

"400": {

"description": "Bad Request",

"content": {

"\*/\*": {

"schema": {

"type": "object",

"additionalProperties": {

"type": "string"

}

}

}

}

},

"200": {

"description": "OK"

}

}

}

}

},

"components": {

"schemas": {

"Characteristics": {

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64"

},

"time": {

"type": "string",

"format": "date-time"

},

"usesNum": {

"type": "integer",

"format": "int32"

},

"averageScore": {

"maximum": 10,

"minimum": 0,

"type": "number",

"format": "double"

},

"effectiveness": {

"maximum": 15,

"minimum": 0,

"type": "number",

"format": "double"

},

"techSolution": {

"$ref": "#/components/schemas/TechSolution"

}

}

},

"GrantedAuthority": {

"type": "object",

"properties": {

"authority": {

"type": "string"

}

}

},

"Score": {

"type": "object",

"properties": {

"id": {

"$ref": "#/components/schemas/ScoreId"

},

"user": {

"$ref": "#/components/schemas/User"

},

"techSolution": {

"$ref": "#/components/schemas/TechSolution"

},

"scoreValue": {

"maximum": 10,

"minimum": 1,

"type": "integer",

"format": "int32"

}

}

},

"ScoreId": {

"type": "object",

"properties": {

"userId": {

"type": "integer",

"format": "int64"

},

"techSolutionId": {

"type": "integer",

"format": "int64"

}

}

},

"TechSolution": {

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64"

},

"name": {

"maxLength": 100,

"minLength": 1,

"type": "string"

},

"documentationUrl": {

"maxLength": 200,

"minLength": 0,

"pattern": "^[a-zA-ZА-я0-9!@#$%^&\*()\_+\\-=\\[\\]{};':\"\\\\|,.\u003C\u003E\\/?]\*$",

"type": "string"

},

"currentEffectiveness": {

"maximum": 15,

"minimum": 0,

"type": "number",

"format": "double"

},

"status": {

"type": "string",

"enum": [

"MOVED\_DOWN",

"MOVED\_UP",

"NEW"

]

},

"category": {

"type": "string",

"enum": [

"LANGUAGES",

"TOOLS",

"TECHNIQUES",

"PLATFORMS"

]

},

"scores": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Score"

}

},

"characteristics": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Characteristics"

}

},

"lastEffectivenessUpdate": {

"type": "string",

"format": "date-time"

}

}

},

"User": {

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64"

},

"login": {

"maxLength": 50,

"minLength": 4,

"type": "string"

},

"password": {

"type": "string"

},

"role": {

"type": "string",

"enum": [

"ADMIN",

"VOTER"

]

},

"scores": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Score"

}

},

"enabled": {

"type": "boolean"

},

"accountNonExpired": {

"type": "boolean"

},

"credentialsNonExpired": {

"type": "boolean"

},

"accountNonLocked": {

"type": "boolean"

},

"username": {

"type": "string"

},

"authorities": {

"type": "array",

"items": {

"$ref": "#/components/schemas/GrantedAuthority"

}

}

}

},

"TechSolutionDTO": {

"required": [

"category",

"name"

],

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64"

},

"name": {

"maxLength": 100,

"minLength": 1,

"type": "string"

},

"documentationUrl": {

"maxLength": 200,

"minLength": 0,

"pattern": "^[a-zA-ZА-я0-9!@#$%^&\*()\_+\\-=\\[\\]{};':\"\\\\|,.\u003C\u003E\\/?]\*$",

"type": "string"

},

"category": {

"type": "string",

"enum": [

"LANGUAGES",

"TOOLS",

"TECHNIQUES",

"PLATFORMS"

]

}

}

},

"AddScoreRequest": {

"required": [

"techId"

],

"type": "object",

"properties": {

"techId": {

"type": "integer",

"format": "int64"

},

"scoreValue": {

"maximum": 10,

"minimum": 1,

"type": "integer",

"format": "int32"

}

}

},

"JwtRefreshRequest": {

"required": [

"refreshToken"

],

"type": "object",

"properties": {

"refreshToken": {

"type": "string"

}

}

},

"RegisterRequest": {

"required": [

"login",

"password",

"role"

],

"type": "object",

"properties": {

"login": {

"maxLength": 50,

"minLength": 4,

"pattern": "^[a-zA-Z0-9\_]\*$",

"type": "string"

},

"password": {

"maxLength": 50,

"minLength": 3,

"pattern": "^[a-zA-Z0-9\_]\*$",

"type": "string"

},

"role": {

"type": "string",

"enum": [

"ADMIN",

"VOTER"

]

}

}

},

"LoginRequest": {

"required": [

"login",

"password"

],

"type": "object",

"properties": {

"login": {

"maxLength": 50,

"minLength": 4,

"pattern": "^[a-zA-Z0-9\_]\*$",

"type": "string"

},

"password": {

"maxLength": 50,

"minLength": 3,

"pattern": "^[a-zA-Z0-9\_]\*$",

"type": "string"

}

}

},

"TechSolutionOnRadarDTO": {

"required": [

"id",

"label",

"moved",

"quadrant",

"ring"

],

"type": "object",

"properties": {

"id": {

"type": "integer",

"format": "int64"

},

"link": {

"maxLength": 100,

"minLength": 1,

"pattern": "^[a-zA-ZА-я0-9!@#$%^&\*()\_+\\-=\\[\\]{};':\"\\\\|,.\u003C\u003E\\/?]\*$",

"type": "string"

},

"quadrant": {

"maximum": 3,

"minimum": 0,

"type": "integer",

"format": "int32"

},

"ring": {

"maximum": 3,

"minimum": 0,

"type": "integer",

"format": "int32"

},

"label": {

"maxLength": 200,

"minLength": 2,

"pattern": "^[a-zA-ZА-я0-9!@#$%^&\*()\_+\\-=\\[\\]{};':\"\\\\|,.\u003C\u003E\\/?]\*$",

"type": "string"

},

"active": {

"type": "boolean"

},

"moved": {

"maximum": 2,

"minimum": -1,

"type": "integer",

"format": "int32"

}

}

},

"TechSolutionOnRadarResponse": {

"type": "object",

"properties": {

"entries": {

"type": "array",

"items": {

"$ref": "#/components/schemas/TechSolutionOnRadarDTO"

}

}

}

},

"ScoreDTO": {

"type": "object",

"properties": {

"techSolution": {

"$ref": "#/components/schemas/TechSolutionDTO"

},

"scoreValue": {

"type": "integer",

"format": "int32"

}

}

},

"DeleteScoreRequest": {

"required": [

"techId"

],

"type": "object",

"properties": {

"techId": {

"type": "integer",

"format": "int64"

}

}

}

},

"securitySchemes": {

"bearerAuth": {

"type": "http",

"scheme": "bearer",

"bearerFormat": "JWT"

}

}

}

}

# 6.1. Описание API

Общее описание

API представляет собой Case Management Service, который предоставляет интерфейсы для управления технологическими решениями, оценками и авторизацией пользователей. API использует спецификацию OpenAPI 3.0 и поддерживает стандартные методы HTTP (GET, POST, PUT, DELETE). Взаимодействие с API осуществляется через различные контроллеры, каждый из которых отвечает за определенные функции.

Сервер

API работает на локальном сервере, доступном по адресу:

* http://localhost:8080

Авторизация

Для доступа к большинству методов API требуется авторизация. Входные данные передаются через заголовки HTTP, в частности, через токен авторизации. Методы авторизации находятся в контроллере auth-controller.

Контроллеры и методы API

1. **tech-solution-controller**
   * **PUT /api/tech/updateTechSolution  
     Обновляет существующее технологическое решение.**
   * **POST /api/tech/updateEffectivenessOnClick**Обновляет эффективность решения при нажатии.
   * **POST /api/tech/createSolution**Создает новое технологическое решение.
   * **GET /api/tech/getAllSolutions**Возвращает список всех технологических решений.
   * **DELETE /api/tech/delete/{id}**Удаляет технологическое решение по идентификатору.
2. **score-controller**
   * **POST /api/score/addOrUpdate**Добавляет новую оценку или обновляет существующую.
   * **GET /api/score/getScores**Возвращает список всех оценок.
   * **GET /api/score/getScoresByTechSolutionsCategory**Возвращает оценки по категории технологических решений.
   * **GET /api/score/getNotScoredTechSolutionsByCategory/{category}**Возвращает технологические решения без оценок в указанной категории.
   * **DELETE /api/score/delete**Удаляет оценку.
3. **auth-controller**
   * **POST /api/auth/token**Получение токена авторизации.
   * **POST /api/auth/register**Регистрация нового пользователя.
   * **POST /api/auth/refresh**Обновление токена авторизации.
   * **POST /api/auth/login**Аутентификация пользователя и получение токена.
4. **statistic-controller**
   * **GET /api/statistic/year**Получение статистики за год.
   * **GET /api/statistic/score/distribution/{tech\_id}**Получение распределения оценок для указанного технологического решения.
   * **GET /api/statistic/month**Получение статистики за месяц.
   * **GET /api/statistic/characteristics**Запись характеристик технологического решения в базу данных вручную.

Схемы данных (Schemas)

API использует несколько схем данных для передачи и обработки информации:

* **Characteristics**: Содержит информацию о характеристиках технологического решения.
* **GrantedAuthority**: Информация о предоставленных правах доступа.
* **Score**: Схема оценки.
* **ScoreId**: Идентификатор оценки.
* **TechSolution**: Схема для технологических решений.
* **User**: Информация о пользователе.
* **TechSolutionDTO**: DTO для технологических решений.
* **AddScoreRequest**: Запрос на добавление оценки.
* **JwtRefreshRequest**: Запрос на обновление JWT-токена.
* **RegisterRequest**: Запрос на регистрацию.
* **LoginRequest**: Запрос на вход.
* **TechSolutionOnRadarDTO**: DTO для отображения решения на радаре.
* **TechSolutionOnRadarResponse**: Ответ с информацией о решении на радаре.
* **ScoreDTO**: DTO для оценки.
* **DeleteScoreRequest**: Запрос на удаление оценки.

# 6.2. Описание внешних интеграций

1. **Авторизация во внешней системе:**
   * Авторизация осуществляется через JWT токен, который можно получить, используя /api/auth/login или /api/auth/register. Для получения нового токена обновления (refreshToken), используется /api/auth/refresh.
   * Для работы с этими токенами необходимо иметь доступ к системе авторизации, которая генерирует и проверяет JWT токены. В зависимости от реализации, могут потребоваться SSL-сертификаты для защиты передачи данных (если API работает через HTTPS).

# 6.3. Маппинг полей на UI к полям АРІ

Вот таблица, которая описывает маппинг полей пользовательского интерфейса (UI) с соответствующими полями API:

| UI Поле | API Поле | Описание |
| --- | --- | --- |
| Название | name | Название технического решения. |
| URL документации | documentationUrl | URL-адрес, указывающий на документацию технического решения. |
| Количество текущих использований | currentUsesNum | Количество текущих использований решения. |
| Средняя оценка | currentAverageScore | Текущая средняя оценка решения. |
| Эффективность | currentEffectiveness | Текущая оценка эффективности решения. |
| Статус | status | Статус решения (например, активен/неактивен). |
| Категория | category | Категория решения (например, LANGUAGES, TOOLS). |
| ID решения | id | Уникальный идентификатор технического решения. |
| Оценка пользователя | scoreValue | Значение оценки, выставленное пользователем. |
| ID пользователя | userId | Уникальный идентификатор пользователя. |
| ID технического решения | techSolutionId | Уникальный идентификатор технического решения для оценки. |
| Токен обновления | refreshToken | Токен для обновления сессии пользователя. |
| Токен доступа | accessToken | Токен доступа для аутентификации пользователя. |
| Электронная почта | email | Электронная почта пользователя для регистрации/входа. |
| Пароль | password | Пароль пользователя для регистрации/входа. |

Эта таблица поможет связать поля, используемые в пользовательском интерфейсе, с соответствующими полями, ожидаемыми API, обеспечивая корректное взаимодействие между фронтендом и бэкендом.

# 6.4. Возможные виды ошибок, их коды и сообщения

Ниже представлено описание ошибок, основанное на спецификации.

Создание решения (/api/tech/createSolution):

* **400 Bad Request**: Возникает, если в запросе отсутствуют обязательные поля, такие как name, type, description, или если поля имеют неверный формат.
* **401 Unauthorized**: Вызывается, если запрос выполняется без авторизации или токен авторизации недействителен.
* **403 Forbidden**: Указывает на то, что у пользователя недостаточно прав для выполнения операции.
* **404 Not Found**: Возникает, если ресурс (например, решение) не найден в системе.
* **422 Unprocessable Entity**: Ошибка валидации данных запроса, например, неверный формат даты или URL.

Редактирование решения (/api/tech/editSolution):

* **400 Bad Request**: Возникает при отсутствии обязательных полей или их неверном заполнении.
* **401 Unauthorized**: Происходит, если пользователь не авторизован.
* **403 Forbidden**: У пользователя нет прав на редактирование данного решения.
* **404 Not Found**: Решение с указанным ID не найдено в системе.
* **422 Unprocessable Entity**: Ошибка валидации входных данных, например, неправильный формат даты.

Получение решения по ID (/api/tech/getSolutionById):

* **401 Unauthorized**: Запрос выполнен без авторизации.
* **403 Forbidden**: Доступ к запрашиваемому ресурсу запрещен.
* **404 Not Found**: Решение с указанным ID не найдено.

Добавление или обновление оценки пользователя (/api/user/addOrUpdateUserScore):

* **400 Bad Request**: Возникает, если в запросе отсутствует поле score или оно имеет неверный формат.
* **401 Unauthorized**: Происходит, если пользователь не авторизован.
* **403 Forbidden**: У пользователя нет прав на обновление оценки.
* **404 Not Found**: Пользователь с указанным ID не найден.
* **422 Unprocessable Entity**: Оценка выходит за допустимые границы, например, не входит в диапазон от 1 до 10.

Получение оценки пользователя (/api/user/getUserScore):

* **401 Unauthorized**: Запрос выполнен без авторизации.
* **403 Forbidden**: Доступ к информации запрещен.
* **404 Not Found**: Пользователь с указанным ID не найден.

Удаление решения (/api/tech/deleteSolution):

* **401 Unauthorized**: Пользователь не авторизован.
* **403 Forbidden**: У пользователя нет прав на удаление решения.
* **404 Not Found**: Решение с указанным ID не найдено.

Поиск решений (/api/tech/searchSolutions):

* **400 Bad Request**: Запрос содержит недопустимые параметры поиска.
* **401 Unauthorized**: Запрос выполнен без авторизации.

Эти описания помогают понять, какие ошибки могут возникнуть при работе с API и как они связаны с различными аспектами запроса.

# 6.5. Описание свойств UI полей: редактируемость, видимость, обязательность

| Поле | Редактируемость  (Editability) | Видимость  (Visibility) | Обязательность  (Mandatory) |
| --- | --- | --- | --- |
| Имя | Editable | Visible | Required |
| Email | Editable | Visible | Required |
| Дата рождения | Read-Only | Visible | Optional |
| ID пользователя | Read-Only | Hidden | Optional |
| Тип пользователя | Editable | Conditional | Conditional Required |
| Комментарий | Editable | Visible | Optional |