**МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВОСИБИРСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«НОВОСИБИРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

СПЕЦИАЛЬНОСТЬ 09.02.03 (230115)

ПРОГРАММИРОВАНИЕ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

**КУРСОВАЯ РАБОТА**

**по теме: Разработка рейтинговой системы студентов колледжа “ТопНПК”**

Студент: Чепилян Семён

Группа: 120 ПКС

Руководитель: Кондюрин В. А.

г. Новосибирск

2024 г.

Оглавление

[Введение 3](#_Toc156990290)

[Изменить по Требованиям к ДП Стр. 17 1.Основная (теоретическая) часть 4](#_Toc156990291)

[1.1 Обзор известных методов решения поставленных задач 5](#_Toc156990292)

[1.2 Описание метода решения поставленной задачи (если необходимо, привести расчеты) 6](#_Toc156990293)

[2.Практическая часть 7](#_Toc156990294)

[Обоснование выбора языка программирования. 9](#_Toc156990295)

[Обоснование выбора интегрированной среды разработки. 9](#_Toc156990296)

[Структура программы (привести все необходимые схемы и диаграммы с подробным описанием). 9](#_Toc156990297)

[Организация данных и внутреннего интерфейса программы. 9](#_Toc156990298)

[Интерфейс пользователя. 9](#_Toc156990299)

[Отладка и тестирование (описать как проводилась тестирование и какие ошибки были найдены и исправлены) 9](#_Toc156990300)

[Меры информационной безопасности (описать используемые средства защиты данных). 9](#_Toc156990301)

[Инструкция пользователя 9](#_Toc156990302)

[3.Экономическая часть 9](#_Toc156990303)

[1. Требования 10](#_Toc156990304)

[2. Архитектура 12](#_Toc156990305)

[3. Стек-технологий 13](#_Toc156990306)

[4. Бизнес-процесс 15](#_Toc156990307)

[5. Диаграмма Прецедентов 16](#_Toc156990308)

[6. Диаграмма последовательностей 17](#_Toc156990309)

[7. ER-диаграмма 18](#_Toc156990310)

[8. Бизнес-план 19](#_Toc156990311)

[**1. Целевая аудитория** 19](#_Toc156990312)

[**2. Резюме** 19](#_Toc156990313)

[**3. Этапы реализации проекта** 20](#_Toc156990314)

[**4. Характеристика объекта** 20](#_Toc156990315)

[**5. План маркетинга** 21](#_Toc156990316)

[**6. Технико-экономические данные оборудования** 21](#_Toc156990317)

[**7. Оценка риска** 22](#_Toc156990318)

[**8. Выводы** 25](#_Toc156990319)

[Заключение 26](#_Toc156990320)

[Список литературы 29](#_Toc156990321)

# Введение

В современном образовательном мире, характеризующемся растущей конкуренцией и стремительными изменениями, эффективное управление учебным процессом и оценка академических достижений студентов становятся ключевыми компонентами обеспечения качества образования. Рейтинговые системы в учебных заведениях становятся неотъемлемой частью этого процесса, предоставляя средство для объективной оценки успеваемости студентов и стимулируя их к активной учебной деятельности.

**Рейтинговая система –** это совокупность правил, методических указаний и соответствующего математического аппарата, реализованного в программном комплексе, обеспечивающем обработку информации, как по количественным, так и по качественным показателям индивидуальной учебной деятельности студентов, позволяющем присвоить персональный рейтинг каждому студенту в разрезе любой дисциплины, любого вида занятий, а также обобщению ряда дисциплин. [1]

В современных колледжах отмечается отсутствие структурированных и эффективных механизмов для оценки активности и достижений студентов во внеурочной деятельности. Эта недостаточная система оценки активностей становится причиной ряда проблем:

1. **Неудовлетворительное отслеживание учебных достижений:** традиционные методы фиксации успехов в учебе сосредотачиваются главным образом на формальных оценках, упуская из виду активность студента в различных внеурочных сферах.
2. **Недостаточное признание дополнительных достижений:** без системы оценки внеурочных активностей студенты, вклад которых выходит за рамки учебных предметов, не получают должного признания за свои дополнительные усилия и таланты.
3. **Отсутствие мотивации к участию во внеурочной деятельности:** студенты, не видящие прямой связи между своей активностью во внеурочных сферах и их общим успехом, могут терять мотивацию к участию в дополнительных мероприятиях.
4. **Ограниченные возможности развития лидерских качеств:** без системы оценки внеурочных достижений, учебное заведение упускает возможность стимулировать студентов к развитию лидерских навыков и профессиональных качеств вне пределов учебных аудиторий.
5. **Несправедливость в распределении ресурсов:** отсутствие объективной системы оценки внеурочных активностей может привести к несправедливому распределению стипендий, наград и других ресурсов, ориентированных на студентов, проявляющих активность в различных сферах.

Решение этих проблем требует внедрения эффективной и справедливой рейтинговой системы, которая учитывает активность и достижения студентов во внеурочной деятельности, способствуя их полноценному развитию и мотивации к дополнительной активности.

**Цель исследования:** Целью данной работы является создание инновационной и адаптивной рейтинговой системы, способствующей улучшению образовательного процесса и повышению мотивации студентов. Разработка такой системы будет способствовать повышению конкурентоспособности Новосибирского политехнического колледжа и обеспечит более точную оценку достижений студентов.

**Задачи исследования:**

1. Провести анализ существующих рейтинговых систем в образовании для выявления наилучших практик и подходов к их проектированию.
2. Выявить критерии оценки и параметры, которые следует учесть при разработке рейтинговой системы для колледжа
3. Разработать модель рейтинговой системы, которая учитывает особенности учебного заведения и способствует справедливой оценке студенческих достижений.
4. Определить методы сбора и анализа данных для эффективной работы рейтинговой системы.
5. Разработать рейтинговую систему и базу данных к ней.
6. Протестировать разработанную систему на выборке студентов и провести анализ ее эффективности.
7. Предложить рекомендации по внедрению и дальнейшему совершенствованию рейтинговой системы.

Гипотеза: Внедрение рейтинговой системы "ТопНПК" в Новосибирском политехническом колледже способствует более объективной оценке достижений студентов, мотивации к активной деятельности и, как следствие, улучшению качества образования в учебном заведении, а также, повышению общего рейтинга и престижа колледжа.

Изменить по Требованиям к ДП Стр. 17

1. **Теоретическая часть**

## **Обзор известных методов решения поставленных задач**

В современном мире существует обширное разнообразие методов и подходов к решению различных задач, связанных с оценкой студенческих успехов и созданием эффективных рейтинговых систем. С учетом динамичного развития образования и технологических изменений, исследователи и практики по всему миру стремятся находить инновационные решения, способствующие улучшению образовательного процесса и мотивации студентов. Я рассмотрю как традиционные, так и современные подходы, чтобы выявить лучшие практики и определить, как можно подходить к созданию эффективной системы для нашего колледжа.

**1. Анализ существующих рейтинговых систем в образовании.**

Для выполнения этой задачи можно провести обширный анализ различных рейтинговых систем в образовании. Методы включают:

* **Литературный обзор:** Изучение академических и практических исследований, касающихся различных рейтинговых систем в образовании.
* **Изучение опыта других учебных заведений:** Анализ успешных кейсов применения рейтинговых систем в других колледжах и университетах.
* **Сравнительный анализ:** Сопоставление преимуществ и недостатков различных систем с целью выделения наилучших практик.

**2. Выявление критериев оценки и параметров.**

Определение ключевых критериев оценки и параметров, соответствующих специфике колледжа. Методы включают:

* **Консультации с преподавателями и администрацией:** Сбор мнений и предложений от профессионалов в образовании.
* **Анализ образовательной программы:** Определение тех критериев, которые наилучшим образом соответствуют учебному плану колледжа.

**3. Разработка модели рейтинговой системы.**

Создание индивидуализированной модели рейтинговой системы с учетом особенностей колледжа. Методы включают:

* **Участие экспертов:** Привлечение экспертов в образовании для разработки критериев и методов оценки.
* **Использование инструментов моделирования:** Применение современных технологий для визуализации и тестирования модели.

**4. Определение методов сбора и анализа данных.**

Разработка эффективных методов сбора и анализа данных для поддержки работы рейтинговой системы. Методы включают:

* **Внедрение автоматизированных систем:** Использование электронных средств для сбора и хранения данных.
* **Обучение персонала:** Проведение обучения сотрудников для правильного сбора и анализа данных.

**5. Разработка рейтинговой системы и базы данных.**

Создание самой рейтинговой системы и ее базы данных. Методы включают:

* **Программирование и разработка:** Использование языков программирования и современных технологий для создания системы.
* **Тестирование безопасности:** Проведение тестирования на безопасность базы данных и системы в целом.

**6. Тестирование и анализ эффективности.**

Проведение тестирования разработанной системы на выборке студентов и анализ ее эффективности. Методы включают:

* **Пилотное тестирование:** Тестирование системы на ограниченной группе пользователей для выявления потенциальных проблем.
* **Сравнительный анализ с предыдущими системами:** Сопоставление результатов с существующими рейтинговыми системами.

**7. Предложение рекомендаций по внедрению и совершенствованию.**

Подготовка практических рекомендаций для успешного внедрения системы и ее последующего совершенствования. Методы включают:

* **Сбор обратной связи:** Проведение опросов и интервью для сбора мнений пользователей.
* **Корректировка на основе результатов:** Внесение изменений в систему с учетом обратной связи для повышения ее эффективности.

## **Описание метода решения поставленной задачи**

В разработке рейтинговой системы для студентов колледжа, применяется современный подход, основанный на использовании языка программирования Python и фреймворка Django. Этот метод предоставляет эффективные инструменты для разработки веб-приложений, а также включает широкий стек технологий, обеспечивающих надежность, безопасность и гибкость системы

**1. Анализ требований и концептуализация:**

Первоначально проведем встречи с представителями колледжа, преподавателями и студентами для выявления конкретных требований к рейтинговой системе. Определим основные критерии оценки, параметры, которые необходимо учесть, и функциональные особенности системы.

**2. Проектирование базы данных:**

На основе выявленных требований разработаем структуру базы данных в PostgreSQL. Определим таблицы для хранения информации о студентах, предметах, оценках, активностях и других необходимых данных. Учтем связи между таблицами для эффективного хранения и запроса данных.

**3. Разработка модели Django:**

Создадим модели Django, отражающие структуру базы данных. Используем ORM (Object-Relational Mapping) Django для удобства взаимодействия с базой данных. Определим методы и функциональность для работы с данными, такие как добавление студентов, выставление оценок, учет активностей и т.д.

**4. Разработка бизнес-логики и API:**

Определим бизнес-логику системы, включая правила расчета рейтинга, присвоение баллов различным критериям и оценкам. Разработаем API с использованием Django REST framework для взаимодействия с системой через HTTP-запросы, что обеспечит легкость интеграции с другими приложениями.

**5. Создание пользовательского интерфейса:**

Используем Django шаблоны, HTML, CSS и JavaScript для создания пользовательского интерфейса. Реализуем страницы для ввода и отображения данных о студентах, их успеваемости и рейтинге. Обеспечим удобную навигацию и визуализацию данных.

**6. Реализация аутентификации и авторизации:**

Внедрим систему аутентификации Django для безопасного доступа к рейтинговой системе. Настроим различные уровни доступа (роли) для преподавателей, студентов и администраторов.

**7. Тестирование и отладка:**

Проведем тестирование каждого модуля системы, включая базу данных, бизнес-логику, API и интерфейс. Отладим возможные ошибки и аномалии в системе.

**8. Документирование:**

Создадим документацию, описывающую структуру базы данных, API-методы, бизнес-логику и особенности пользовательского интерфейса. Это обеспечит легкость поддержки и дальнейшего развития системы.

**9. Разворачивание и поддержка:**

Развернем приложение на хостинге Beget, обеспечивая безопасность данных. Внедрим Gunicorn в качестве WSGI-сервера. Предоставим необходимые настройки для эффективного функционирования системы. Организуем процессы обновления и поддержки.

**10. Интеграция дополнительных инструментов:**

Интегрируем дополнительные библиотеки Django, такие как Django REST framework для API и Django Celery для управления асинхронными задачами. Это обеспечит расширенные функциональные возможности системы.

**Важные моменты:**

**Git и GitHub:** Используем систему контроля версий Git с репозиторием на GitHub для эффективной работы в команде и отслеживания изменений.

**Обеспечение безопасности:** При разработке системы будем придерживаться лучших практик по обеспечению безопасности, включая защиту от инъекций, аутентификацию и авторизацию.

## **Обоснование выбора программных средств**

При разработке рейтинговой системы для колледжа, выбор программных средств играет ключевую роль в обеспечении эффективности, надежности и удобства использования системы. Это обоснование направлено на разъяснение принятых решений по выбору технологического стека и его соответствия задачам проекта.

Под стеком технологий понимается совокупность технологических принципов, компетенций и решений, используемых в разработке продукта (веб-сайта, программы для ПК, мобильного приложения) и обеспечивающих эффективное его использование на практике. [[9]](https://www.diasoft.ru/about/publications/20724/?ysclid=lpozu1183755174414)

**Требования к программным средствам:**

**Гибкость и быстродействие:** Необходимость в быстром и гибком инструментарии для разработки, управления данными и обеспечения высокой производительности системы.

**Безопасность и защита данных:** Защита персональной информации студентов и обеспечение безопасности данных системы.

**Удобство разработки и поддержки:** Простота в разработке, а также удобство поддержки и дальнейшего развития приложения.

**Интеграция с современными технологиями:** Возможность интеграции с современными библиотеками и фреймворками для расширения функциональности системы.

Для удовлетворения этих требований был выбран современный и мощный стек технологий:

1. **Языки программирования:**
   * **Python.** Выбор языка программирования Python обусловлен его универсальностью, легкостью в использовании и выразительностью кода. Python предоставляет широкий набор библиотек и фреймворков, что существенно упрощает разработку веб-приложений. Его активное сообщество и богатая экосистема делают Python идеальным выбором для быстрой и эффективной реализации рейтинговой системы для студентов колледжа.
   * **JavaScript.** Выбор языка обоснован его важной ролью в разработке интерактивных и динамических пользовательских интерфейсов веб-приложений. JavaScript позволяет создавать более отзывчивые и интуитивно понятные веб-страницы, обеспечивая пользователям более удобный опыт взаимодействия с системой.
2. **Фреймворк:**
   * **Django.** Фреймворк Django был выбран в качестве основы для разработки рейтинговой системы в связи с его выдающимися возможностями для построения веб-приложений. Django, изначально созданный для обеспечения быстрого и эффективного проектирования веб-приложений, стал оптимальным решением для создания надежной и масштабируемой рейтинговой системы.
3. **База данных:**
   * **PostgreSQL.** Выбор PostgreSQL в роли системы управления базами данных для рейтинговой системы колледжа обоснован его выдающейся надежностью, высокой производительностью и расширенными возможностями. Он обладает гибкостью в работе с различными типами данных, что значительно упрощает работу.
4. **Интерфейс пользователя (Frontend):**
   * **HTML, CSS.** Включение HTML и CSS в технологический стек обусловлено необходимостью эффективного формирования и оформления пользовательского интерфейса, обеспечивая ясную структуру и визуальную привлекательность веб-приложения с самыми современными подходами в разработке.
5. **Система управления версиями:**
   * **GitHub.** Интеграция с GitHub в технологическом стеке необходима для эффективного управления версиями кода, а также для сохранения целостности и чистоты проекта.
6. **Веб-сервер:**
   * **Gunicorn.** Gunicorn выбран в технологическом стеке для обслуживания веб-приложения. Этот WSGI-сервер обеспечивает высокую производительность и стабильность, что делает его оптимальным выбором для развертывания рейтинговой системы. Он заранее предустановлен в фреймворк Django, что делает работу еще более удобной.
7. **Хостинг:**
   * **Beget.** Хостинг ообеспечивает безопасность данных, а также сервера данного хостинга находятся на территории России. Есть возможность разместить сайт бесплатно. Beget работает надёжно и проверен мною на других проектах.
8. **Дополнительные библиотеки и пакеты Django:**
   * **Django REST framework:** для возможностии создания API для взаимодействия с рейтинговой системой.
   * **Django Celery:** для управления асинхронными задачами, например, отправкой уведомлений или обновлением данных.

## **Описание программных средств и методик разработки**

Осмысленное рассмотрение определений программных средств, языков программирования и выбранных методик разработки является неотъемлемой частью освоения основных концепций. В этом контексте происходит взгляд на сущность каждого инструмента и метода, выявление их ключевых характеристик и понимание того, как они взаимодействуют в контексте разработки рейтинговой системы. Разбор этих определений становится фундаментальным шагом на пути к глубокому владению технологическим стеком проекта, что, в свою очередь, является необходимым условием для успешной реализации поставленных целей и задач.

**Программные средства разработки:**

* **Python —** это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Он ориентирован на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Python является полностью объектно-ориентированным языком. [2]
* **JavaScript** — это язык программирования, который в первую очередь применяют в вебе. С его помощью сайты делают интерактивными: добавляют всплывающие окна, анимацию, кнопки лайков и формы для отправки информации. Его ещё называют главным языком фронтенда — «лицевой» стороны сайта, с которой взаимодействуют пользователи. [3]
* **Django** — это высокоуровневый Python веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-сайты. Он бесплатный и с открытым исходным кодом, имеет растущее и активное сообщество, отличную документацию и множество вариантов как бесплатной, так и платной поддержки. [4]
* **PostgreSQL** — это свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом. «Объектно-реляционная» означает, что PostgreSQL поддерживает концепции, присущие как реляционным базам данных, так и объектно-ориентированным языкам программирования (объекты, классы, наследование и другие). [5]
* **HTML** — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео. [6]
* **CSS** — язык описания внешней страницы. Он отвечает за визуальную красоту и оформление сайта. [7]
* **GitHub** — это веб-сервис, который основан на системе Git. Это такая социальная сеть для разработчиков, которая помогает удобно вести коллективную разработку IT-проектов. Здесь можно публиковать и редактировать свой код, комментировать чужие наработки, следить за новостями других пользователей. [8]
* **Gunicorn** — это Application-сервер для запуска Web-приложений написанных на Python. Основная его задача — это работа в режиме демона и поддержка постоянной работы Web-приложений. [9]
* **Beget** — международный хостинг-провайдер, аккредитованный национальный и международный регистратор доменных имён. [10]
* **Django** **REST** **framework** (DRF) — это мощный и гибкий инструмент для создания Web API на основе Django. Он предоставляет удобные средства для создания RESTful API, поддерживает аутентификацию, авторизацию, сериализацию, валидацию и другие функции. [11]
* **Celery** – это инструмент для обработки задач в фоновом режиме в Django-приложениях. Он позволяет разделить основной поток выполнения от обработки тяжелых задач, повышая производительность и отзывчивость приложения. Celery поддерживает очереди задач, отложенное выполнение, приоритеты, планирование и мониторинг выполнения задач. [12]

# Практическая часть

# **Выбор аппаратной части**

Для обеспечения стабильной и эффективной работы программных средств, а также поддержки требуемого технологического стека, неотъемлемым компонентом является правильно подобранная аппаратная часть. Ключевым моментом при выборе аппаратуры является обеспечение оптимальной производительности, масштабируемости и надежности системы. В данном контексте, особое внимание уделяется адаптации аппаратной инфраструктуры под требования, выдвигаемые используемыми программными средствами, а также спецификациями технологического стека.

Таблица 1. Аппаратная часть (рекомендуемые требования)

|  |  |  |  |
| --- | --- | --- | --- |
| **Компонент** | **Минимальные Требования** | **Рекомендуемые Требования** | **Рекомендуемые Бренды** |
| Процессор (CPU) | Двухъядерный, 2.0 ГГц | Четырехъядерный, 3.0 ГГц | Intel Core i5, AMD Ryzen 5 |
| Оперативная Память (RAM) | 8 ГБ RAM | 16 ГБ RAM | DDR4 2666 МГц и выше |
| Хранилище данных | SSD, 120 ГБ свободного места | SSD, 240 ГБ свободного места | Samsung, Crucial, Western Digital |
| Графическая карта (GPU) | Интегрированная или дискретная, поддерживающая Full HD разрешение | Дискретная, Full HD, NVIDIA или AMD | NVIDIA GeForce GTX или RTX, AMD Radeon |
| Сетевая карта | 1 Гбит/сеть | 1 Гбит/сеть | Intel, TP-Link, ASUS |
| Материнская Плата | Совместимая с процессором, поддерживающая нужное количество слотов и портов | ATX или Micro-ATX форм-фактор, поддерживающая нужные возможности | ASUS, MSI, Gigabyte, ASRock |
| Блок Питания | Мощность 450 Ватт | Мощность 650 Ватт | EVGA, Corsair, Seasonic, Thermaltake |
| Корпус | ATX или Micro-ATX форм-фактор | Вентиляция и простор для хорошей циркуляции воздуха | NZXT, Fractal Design, Corsair, Cooler Master |
| Охлаждение | Воздушное или жидкостное охлаждение, совместимое с процессором | Эффективное охлаждение для поддержки процессора и графической карты | Noctua, be quiet!, Corsair, NZXT |
| Звуковая Карта | Встроенная или дополнительная, в зависимости от потребностей | Дополнительная звуковая карта для профессионального использования | ASUS Xonar, Creative Sound Blaster |
| Монитор | Full HD разрешение, 60 Гц | Quad HD или 4K разрешение, 144 Гц или выше | ASUS, Dell, LG, Acer |
| Клавиатура и Мышь | Беспроводные или проводные, в зависимости от предпочтений | Механическая клавиатура, оптическая мышь | Logitech, Corsair, Razer, SteelSeries |
| Звуковая Система | Встроенные динамики монитора или дополнительные колонки | Дополнительные колонки или наушники с хорошим звуком | Bose, Audioengine, Sennheiser, Logitech |
| Бесперебойник (UPS) | Мощность, достаточная для поддержания системы в работе при сбоях питания | Устройство с автономной батареей и защитой от перенапряжения | APC, CyberPower, Eaton |

При разработке проекта я тщательно подобрал компоненты, учитывая рекомендуемые параметры для обеспечения оптимальной производительности.

Таблица 2. Аппаратная часть моего ПК

|  |  |  |
| --- | --- | --- |
| **Компонент** | **Модель/Описание** | **Цена на 2024 год.** |
| Процессор | AMD Ryzen 5 5600G OEM | 11 999 ₽ |
| Материнская плата | MSI B550M PRO-VDH | 12 499 ₽ |
| Кулер для процессора | ID-COOLING SE-914-XT BASIC | 2 550 ₽ |
| Оперативная память | Оперативная память Goodram Iridium Pro  16Гб 3600МГц | 7 499 ₽ |
| SSD Накопитель | Kingston KC2500, 250 ГБ, M.2 | 3790₽ |
| Жесткий диск | WD Blue, 1 ТБ | 7990₽ |
| Блок питания | Блок питания GIGABYTE AORUS P850W | 11 999 ₽ |
| Корпус | Корпус Cougar Duoface Pro RGB | 8 699 ₽ |
| Монитор | ASUS TUF GAMING Vg279q1r | 16 999 ₽ |
| Мышь | Мышь проводная DEXP Bloodlust | 1 199 ₽ |
| Клавиатура | Клавиатура беспроводная DEXP KW-904BU | 1 099 ₽ |

Мой выбор основан на балансе между производительностью, надежностью и соответствием требованиям проекта, что обеспечивает наилучшие условия для успешной реализации задач.

# **Выбор программной платформы**

Выбор программной платформы представляет собой важный этап в разработке рейтинговой системы. На данном этапе производится настройка и установка основных программных средств, включая операционную систему (ОС), фреймворк Django, интегрированную среду разработки PyCharm, а также необходимые библиотеки и пакеты для разработки веб-приложения. Рассмотрим этапы установки и настройки программных средств.

* + - 1. **Операционная система.**

Таблица 3. Установка/настройка Windows 11 Pro

|  |  |
| --- | --- |
| **Шаг** | **Описание** |
| **Подготовка к установке** | - Загрузите образ операционной системы Windows 11 Pro с официального сайта Microsoft. |
|  | - Или используйте установочный носитель (USB Flash Drive, DVD). |
|  | - Проверьте системные требования Windows 11 Pro. |
| **Запуск установки** | - Подключите установочный носитель к компьютеру. |
|  | - Запустите компьютер и настройте BIOS/UEFI на загрузку с установочного носителя. |
| **Выбор языка и региона** | - Выберите язык, регион, а также параметры времени и валюты. |
| **Начало установки** | - Нажмите "Установить сейчас" и следуйте инструкциям мастера установки. |
| **Лицензионное соглашение** | - Прочитайте и примите лицензионное соглашение Microsoft. |
| **Выбор типа установки** | - Выберите "Пользовательская установка" для чистой установки. |
| **Выбор диска** | - Выберите диск и создайте новый раздел или выберите существующий. |
| **Установка ОС** | - Дождитесь завершения процесса установки. |
| **Настройка параметров** | - Введите информацию о пользователе, включая имя, пароль и вопросы безопасности. |
| **Завершение установки** | - После завершения установки, выполните необходимые настройки, включая подключение к интернету и активацию Windows 11 Pro. |

Теперь компьютер оборудован операционной системой Windows 11 Pro, и можно приступать к установке необходимых программных средств для разработки рейтинговой системы.

* + - 1. **Среда разработки.**

Таблица 3. Установка/настройка PyCharm

|  |  |
| --- | --- |
| **Шаг** | **Описание** |
| **Загрузка установщика** | - Загрузите установщик PyCharm с официального сайта JetBrains. |
| **Запуск установщика** | - Запустите установочный файл PyCharm после его загрузки. |
| **Выбор типа установки** | - Выберите "Установка для всех пользователей", если используется общий компьютер, или "Установка только для меня". |
| **Выбор пути установки** | - Выберите путь, по которому будет установлен PyCharm. |
| **Выбор компонентов** | - Выберите необходимые компоненты для установки. |
| **Настройка ассоциаций файлов** | - Определите, нужно ли связывать файлы с PyCharm. |
| **Создание ярлыка** | - Выберите, нужно ли создавать ярлык на рабочем столе и в меню "Пуск". |
| **Настройка темы и кодировки** | - Выберите тему оформления и кодировку по умолчанию. |
| **Настройка среды запуска** | - Укажите параметры среды запуска (по умолчанию используется встроенная среда). |
| **Завершение установки** | - Дождитесь завершения установки PyCharm и запустите приложение. |

После успешной установки PyCharm, мы получаем удобное и мощное средство для создания, отладки и поддержки рейтинговой системы.

* + - 1. **Фреймворк.**

Таблица 3. Установка/настройка Django

|  |  |
| --- | --- |
| **Шаг** | **Описание** |
| **Установка Django** | - Используя pip, установите Django с помощью команды pip install django. |
| **Создание Django-проекта** | - Выполните команду django-admin startproject projectname, чтобы создать новый проект Django. |
| **Создание Django-приложения** | - Воспользуйтесь командой python manage.py startapp appname, чтобы создать новое приложение Django. |
| **Настройка базы данных** | - В файле settings.py проекта Django определите настройки базы данных, такие как тип и путь к базе данных. |
| **Создание моделей Django** | - Определите модели данных внутри созданного приложения. |
| **Создание миграций** | - Выполните python manage.py makemigrations для создания миграций на основе определенных моделей. |
| **Применение миграций** | - Затем выполните python manage.py migrate для применения миграций и создания схемы базы данных. |
| **Создание Django-админки** | - Определите административные возможности, создав файл admin.py в приложении и зарегистрировав модели. |
| **Создание URL-маршрутов** | - В файле urls.py приложения определите URL-маршруты для обработки различных запросов. |
| **Создание Django-представлений** | - Определите представления для обработки запросов, связанных с вашими URL-маршрутами. |

Успешно завершена установка и настройка необходимых средств разработки для создания рейтинговой системы студентов. В результате:

1. **Операционная Система Windows 11 Pro:** Операционная система была установлена и настроена в соответствии с требованиями.
2. **PyCharm:** Интегрированная среда разработки PyCharm была успешно установлена, что предоставляет удобное и мощное средство для написания, отладки и поддержки кода.
3. **Django:** Веб-фреймворк Django был установлен и настроен для создания веб-приложения, обеспечивая удобную структуру проекта, работу с базой данных и обработку запросов.

Эти шаги обеспечивают необходимую основу для начала активного процесса разработки рейтинговой системы, при этом все инструменты готовы к использованию.

# Обоснование выбора языка программирования.

В рамках данного проекта принято решение использовать два ключевых языка программирования: Python и JavaScript.

1. **Python для серверной части (Django):**
   * **Ясный и читаемый синтаксис:** Python известен своим простым и легко читаемым синтаксисом, что упрощает разработку и поддержку кода. Это особенно важно в контексте создания веб-приложения с использованием Django.
   * **Большое сообщество:** Python имеет активное и большое сообщество разработчиков. Это обеспечивает наличие множества библиотек, фреймворков и ресурсов, что упрощает разработку и обмен опытом.
   * **Мощный фреймворк Django:** Django является мощным и распространенным веб-фреймворком для Python, предоставляя удобные инструменты для работы с базой данных, URL-маршрутизацией, шаблонами и многим другим.
   * **Широкие Возможности Интеграции:** Python может быть легко интегрирован с другими технологиями, что дает свободу выбора в использовании различных библиотек и инструментов.
2. **JavaScript для клиентской части:**
   * **Клиентская Интерактивность:** JavaScript является основным языком для разработки интерактивных пользовательских интерфейсов на стороне клиента. Это позволяет создать динамические и отзывчивые веб-приложения.
   * **Библиотека React (или другие фреймворки):** Использование JavaScript не ограничивается лишь его базовыми возможностями. Фреймворки, такие как React, предоставляют мощные инструменты для создания сложных интерфейсов с использованием компонентов.
   * **Асинхронные Запросы:** JavaScript отлично подходит для работы с асинхронными запросами, что важно при взаимодействии с сервером без перезагрузки страницы (AJAX).
   * **Широкая Поддержка Браузерами:** JavaScript является стандартом для веб-разработки и поддерживается всеми современными браузерами без необходимости установки дополнительных плагинов.

Общее использование Python для серверной части и JavaScript для клиентской части в Django проекте обеспечивает гармоничную интеграцию, позволяя создавать высокопроизводительные и интерактивные веб-приложения.

# 2.4 Обоснование выбора интегрированной среды разработки.

Выбор интегрированной среды разработки (IDE) является ключевым этапом, определяющим процесс создания и поддержки проекта. В данном проекте принято решение использовать PyCharm, и вот почему:

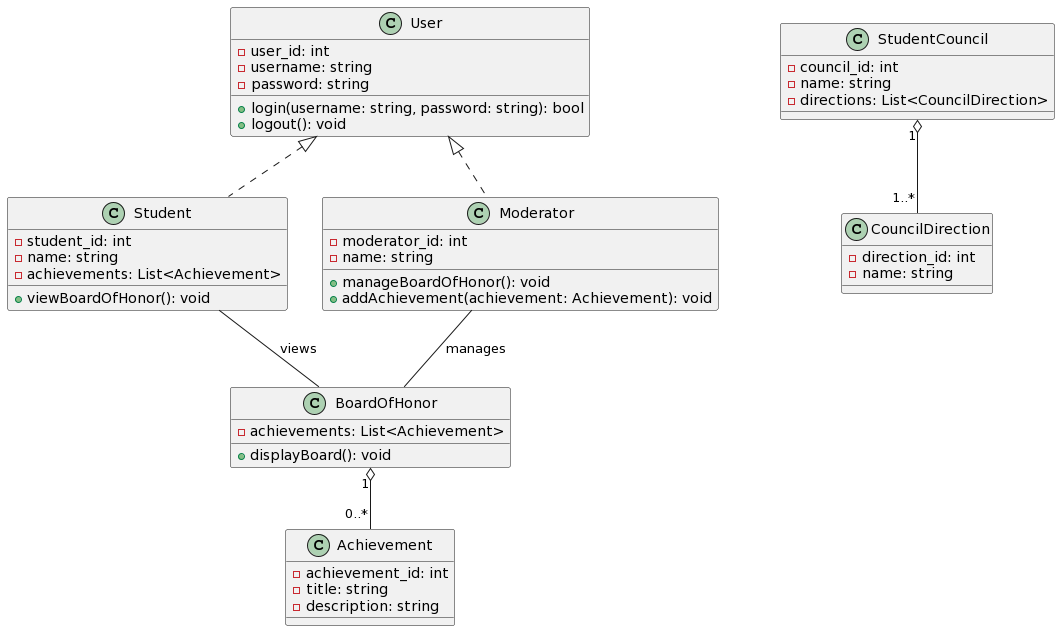
1. **Поддержка Python и JavaScript:**
   * **Python:** PyCharm обеспечивает отличную интеграцию с Python, предоставляя функциональные возможности для написания, отладки и управления кодом.
   * **JavaScript:** PyCharm также эффективно поддерживает разработку на JavaScript, включая функциональности для работы с фреймворками, такими как React.
2. **Удобство и производительность:**
   * **Интуитивный интерфейс:** PyCharm предлагает интуитивный и легко осваиваемый интерфейс, что существенно сокращает время на адаптацию и обучение.
   * **Мощные инструменты:** IDE предоставляет множество мощных инструментов, таких как система контроля версий, автоматическое дополнение кода, отладка и профилирование.
3. **Интеграция с Django:**
   * **Поддержка Django-проектов:** PyCharm предоставляет специальные инструменты для работы с проектами, основанными на Django, упрощая создание и управление веб-приложениями.
4. **Активное сообщество и поддержка:**
   * **Большое сообщество пользователей:** PyCharm имеет широкое и активное сообщество пользователей, что обеспечивает доступ к множеству ресурсов и поддержке со стороны сообщества.
5. **Многоплатформенность:**
   * **Работа на различных ОС:** PyCharm поддерживает многоплатформенность, что позволяет работать на различных операционных системах, включая Windows, macOS и Linux.

Выбор PyCharm обоснован его многофункциональностью, удобством использования, поддержкой ключевых языков программирования проекта и активным сообществом разработчиков. Эта IDE обеспечивает надежное и эффективное средство для разработки и поддержки рейтинговой системы студентов.

# Структура программы

Структура программы включает в себя ряд диаграмм, обеспечивающих качественную работу и полное понимание архитектуры разрабатываемой системы. Каждая из них направлена на детальное описание структуры и взаимодействия компонентов системы, что обеспечивает основу для успешной реализации и поддержки проекта. (добавить архитектуру)

**Диаграмма классов** — структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)) языка моделирования [UML](https://ru.wikipedia.org/wiki/UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей (отношений) между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования. [13]



**Описание диаграммы классов:**

1. **Пользователь (User):**
   * Информация о пользователе (ID, имя, пароль).
   * Методы: вход в систему (login), выход из системы (logout).
2. **Студент (Student):**
   * Информация о студенте (ID, имя).
   * Список достижений.
   * Возможность просмотра доски почета.
3. **Модератор (Moderator):**
   * Информация о модераторе (ID, имя).
   * Возможность управления доской почета.
   * Добавление новых достижений.
4. **Доска почета (BoardOfHonor):**
   * Список достижений, отображаемых на доске почета.
   * Возможность отображения доски.
5. **Достижение (Achievement):**
   * Информация о достижении (ID, название, описание).
6. **Студенческий совет (StudentCouncil):**
   * Информация о студенческом совете (ID, название).
   * Список направлений совета.
7. **Направление совета (CouncilDirection):**
   * Информация о направлении студенческого совета (ID, название).

**Связи:**

* **Пользователь (User)** может быть как **Студентом (Student)**, так и **Модератором (Moderator)**.
* **Студент (Student)** может просматривать доску почета и имеет список своих достижений.
* **Модератор (Moderator)** управляет доской почета и добавляет новые достижения.
* Доска Почета (BoardOfHonor) содержит список достижений.
* Существует отношение "один ко многим" между **Студенческим Советом (StudentCouncil)** и **Направлением Совета (CouncilDirection)**.

**ERD или ER-диаграмма** (также схема «сущность-связь») — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. [[7]](https://www.lucidchart.com/pages/ru/erd-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)



**Для описания ER-диаграммы составим словарь данных:**

Таблица 4. Словарь данных таблицы «Students»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| student\_id | INT | PRIMARY KEY | Уникальный идентификатор каждого студента. |
| first\_name | VARCHAR(50) | NOT NULL | Имя студента, не может быть NULL. |
| last\_name | VARCHAR(50) | NOT NULL | Фамилия студента, не может быть NULL. |
| date\_of\_birth | DATE |  | Дата рождения студента. |
| gender | CHAR(1) |  | Пол студента. |
| email | VARCHAR(100) |  | Email студента. |
| phone\_number | VARCHAR(20) |  | Номер телефона студента. |
| enrollment\_date | DATE | NOT NULL | Дата зачисления студента, не может быть NULL. |

Таблица 4. Словарь данных таблицы «Courses»

|  |  |  |  |
| --- | --- | --- | --- |
| course\_id | INT | PRIMARY KEY | Уникальный идентификатор каждого курса. |
| title | VARCHAR(255) | NOT NULL | Название курса, не может быть NULL. |
| description | TEXT |  | Описание курса. |
| department | VARCHAR(100) |  | Отделение, к которому относится курс. |
| instructor | VARCHAR(100) |  | Имя преподавателя для курса. |
| start\_date | DATE | NOT NULL | Дата начала курса, не может быть NULL. |
| end\_date | DATE | NOT NULL | Дата окончания курса, не может быть NULL. |

Таблица 5. Словарь данных таблицы «Achievements»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| achievement\_id | INT | PRIMARY KEY | Уникальный идентификатор для каждого достижения. |
| title | VARCHAR(255) | NOT NULL | Название достижения, не может быть NULL. |
| description | TEXT |  | Описание достижения. |

Таблица 6. Словарь данных таблицы «StudentCourses»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| student\_id | INT |  | Идентификатор студента. |
| course\_id | INT |  | Идентификатор курса. |
| grade | FLOAT |  | Оценка студента по курсу. |
| enrollment\_date | DATE | NOT NULL | Дата зачисления на курс, не может быть NULL. |

Таблица 7. Словарь данных таблицы «StudentAchievements»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| student\_id | INT |  | Идентификатор студента. |
| achievement\_id | INT |  | Идентификатор достижения. |
| date\_achieved | DATE | NOT NULL | Дата достижения, не может быть NULL. |

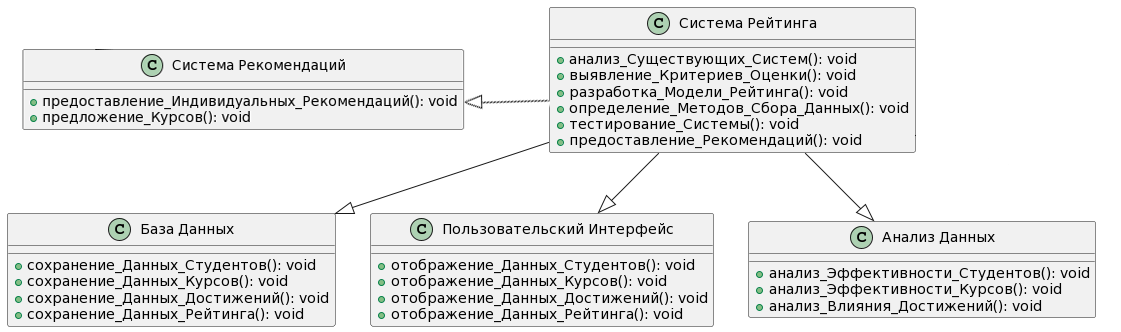
Таблица 8. Словарь данных таблицы «StudentCouncil»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| council\_id | INT | PRIMARY KEY | Уникальный идентификатор студенческого совета. |
| name | VARCHAR(255) | NOT NULL | Название студенческого совета, не может быть NULL. |
| president | VARCHAR(100) |  | Имя президента совета. |
| founding\_date | DATE | NOT NULL | Дата основания совета, не может быть NULL. |

Таблица 9. Словарь данных таблицы «CouncilDirections»

|  |  |  |  |
| --- | --- | --- | --- |
| **Колонка** | **Тип данных** | **Ограничения** | **Описание** |
| direction\_id | INT | PRIMARY KEY | Уникальный идентификатор направления совета. |
| name | VARCHAR(255) | NOT NULL | Название направления совета, не может быть NULL. |
| council\_id | INT |  | Идентификатор студенческого совета. |

**Диаграмма компонентов** — элемент языка моделирования [UML](https://ru.wikipedia.org/wiki/UML), статическая структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), которая показывает разбиение программной системы на структурные [компоненты](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82_(UML)&action=edit&redlink=1) и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать [файлы](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), библиотеки, модули, исполняемые файлы, пакеты и т.п. [14]

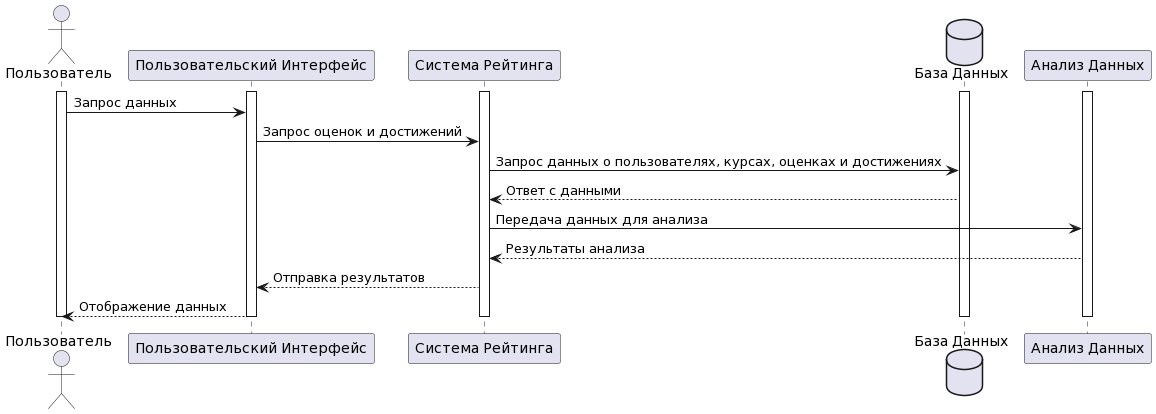


**Описание диаграммы компонентов:**

1. **Система Рейтинга:**
   * анализ\_Существующих\_Систем(): проведение анализа существующих рейтинговых систем в образовании.
   * выявление\_Критериев\_Оценки(): выявление критериев оценки и параметров для разработки рейтинговой системы.
   * разработка\_Модели\_Рейтинга(): создание модели рейтинговой системы с учетом особенностей учебного заведения.
   * определение\_Методов\_Сбора\_Данных(): выбор методов сбора и анализа данных для эффективной работы рейтинговой системы.
   * тестирование\_Системы(): тестирование разработанной системы на выборке студентов.
   * предоставление\_Рекомендаций(): предложение рекомендаций по внедрению и совершенствованию рейтинговой системы.
2. **База данных:**
   * сохранение\_Данных\_Студентов(): хранение данных о студентах.
   * сохранение\_Данных\_Курсов(): хранение данных о курсах.
   * сохранение\_Данных\_Достижений(): хранение данных о достижениях студентов.
   * сохранение\_Данных\_Рейтинга(): хранение данных о рейтинге студентов.
3. **Пользовательский интерфейс:**
   * отображение\_Данных\_Студентов(): отображение информации о студентах.
   * отображение\_Данных\_Курсов(): отображение информации о курсах.
   * отображение\_Данных\_Достижений(): отображение информации о достижениях.
   * отображение\_Данных\_Рейтинга(): отображение рейтинга студентов.
4. **Анализ данных:**
   * анализ\_Эффективности\_Студентов(): проведение анализа эффективности студентов.
   * анализ\_Эффективности\_Курсов(): проведение анализа эффективности курсов.
   * анализ\_Влияния\_Достижений(): анализ влияния достижений на успеваемость студентов.
5. **Система рекомендаций:**
   * предоставление\_Индивидуальных\_Рекомендаций(): предоставление индивидуальных рекомендаций студентам.
   * предложение\_Курсов(): предложение курсов на основе анализа данных.

Взаимосвязи между компонентами позволяют обеспечить целостность и эффективное взаимодействие в рамках создаваемой рейтинговой системы.

**Диаграмма последовательности** — это подвид диаграмм взаимодействия, который позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени. [[6]](https://itonboard.ru/analysis/394-diagramma_posledovatelnosti_sequence_diagrams_uml/?ysclid=lpf8sunzrk739781306)



**Описание диаграммы последовательности:**

1. **Пользователь** (актер):
   * Инициирует запрос данных, например, запрашивает свои оценки и достижения.
2. **Пользовательский Интерфейс (UI)**:
   * Получает запрос от пользователя.
   * Передает запрос системе рейтинга.
3. **Система Рейтинга**:
   * Получает запрос от UI.
   * Инициирует запрос данных к базе данных для получения информации о пользователях, курсах, оценках и достижениях.
   * Получает ответ от базы данных.
   * Передает данные системе анализа данных для дополнительного анализа.
4. **База Данных**:
   * Получает запрос от системы рейтинга.
   * Отправляет ответ с данными, включая информацию о пользователях, курсах, оценках и достижениях.
5. **Анализ Данных**:
   * Получает данные от системы рейтинга для анализа.
   * Выполняет анализ данных.
   * Отправляет результаты анализа обратно в систему рейтинга.
6. **Пользовательский Интерфейс (UI)**:
   * Получает результаты от системы рейтинга.
   * Отображает данные пользователю.

Диаграмма последовательности иллюстрирует весь процесс от запроса пользователя до получения и отображения данных, включая взаимодействие между компонентами системы.

**Диаграмма прецедентов** — диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. [[5]](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2)



переделать

# 2.6 Организация данных и внутреннего интерфейса программы.

# 2.7 Интерфейс пользователя.

# 2.8 Отладка и тестирование (описать как проводилась тестирование и какие ошибки были найдены и исправлены)

# 2.9 Меры информационной безопасности (описать используемые средства защиты данных).

# 2.10 Инструкция пользователя

Этот раздел по значимости является центральным и на его долю должна приходиться большая часть дипломной работы. Он представляет собой аналитическую часть работы и выполняется по материалам, собранным во время преддипломной практики. В практическом разделе приводится подробная информация об исследовании предмета. Осуществляется анализ всех полученных данных, используются различные методы исследования.

В этом разделе решаются поставленные во введении задачи, определяются оптимальные для этого методы. Практические примеры позволяют раскрыть актуальность выбранной темы и обосновать ее. Получив результаты исследования, необходимо сделать на их основании определенные выводы и предложить мероприятия по совершенствованию исследуемого предмета. Это должны быть конкретные предложения с планом проведения таких мероприятий. Наличие и количество таких выводов свидетельствует об актуальности работы. Необходимо также описать перспективы развития.

Рекомендуется следующая последовательность разделов данной части:

# 3. Экономическая часть

# 3.1 Бизнес-план

Целевой аудиторией являются студенты Новосибирского политехнического колледжа колледжа, администрация учебного заведения, и потенциальные работодатели. Каждая из этих групп имеет свои уникальные потребности и ожидания от рейтинговой системы:

* **Студенты:** Они ожидают мотивации, разнообразных поощрений и возможности для развития навыков в рамках своего обучения.
* **Администрация:** Им важны инструменты для эффективного мониторинга академической успеваемости и общей активности студентов, а также инструменты для анализа эффективности образовательных программ.
* **Потенциальные работодатели:** Они заинтересованы в доступе к информации о достижениях студентов, их навыках и подготовке, чтобы более эффективно отбирать перспективных выпускников.

## **2. Резюме**

Я предлагаю создание и внедрение инновационной рейтинговой системы для колледжа с использованием Django технологий. Система нацелена на улучшение мотивации студентов, предоставление им обратной связи для развития навыков и поддержка выбора индивидуального образовательного пути. Система будет включать в себя виртуальную доску почета, мотивационные элементы и индивидуализированный подход к обучению. Проект ориентирован на долгосрочное сотрудничество с вашим колледжем.

## **3. Этапы реализации проекта**

1. **Подготовительный этап:**
   * Изучение потребностей студентов и администрации колледжа.
   * Формирование команды проекта и определение ролей.
2. **Разработка рейтинговой системы:**
   * Проектирование базы данных и API с использованием Django технологий.
   * Создание функциональных элементов системы, включая мотивационные и обратные связи.
3. **Тестирование и оптимизация:**
   * Проведение тестирования системы на пилотной группе студентов.
   * Внесение корректив в систему на основе обратной связи.
4. **Внедрение и обучение:**
   * Запуск рейтинговой системы для всех студентов.
   * Проведение обучающих мероприятий для студентов и администрации.

## **4. Характеристика объекта**

Объектом проекта является создание цифровой рейтинговой системы, интегрированной в учебный процесс вашего колледжа. Основные характеристики включают:

* **Виртуальная доска почета:** Публичное отображение выдающихся студентов.
* **Мотивационные элементы:** Система поощрений и челленджей для стимулирования активности.
* **Новости и жизнь в нашем колледже:** Новостной стенд с главной информацией, достижения нашего колледжа.

## **5. План маркетинга**

* **Цифровой маркетинг:** Реклама в социальных сетях, создание контента для привлечения внимания.
* **Партнерство с колледжем:** Взаимодействие с администрацией для внедрения системы в учебный процесс.
* **Промо-мероприятия:** Организация мероприятий, нацеленных на привлечение внимания студентов.

## **6. Технико-экономические данные оборудования**

* **Серверы:** Закупка и обслуживание выделенных серверов для хранения данных.
* **Программное обеспечение:** Лицензирование Django технологий, адаптированных под требования проекта.

7. Оценка риска

Риски - это потенциальные негативные события, обстоятельства или процессы, которые могут возникнуть и оказать воздействие на достижение целей, выполнение задач или успешное функционирование организации или проекта. Риски могут варьироваться по своей природе, включая финансовые, операционные, стратегические, репутационные и другие аспекты, и могут возникать как из-за внутренних факторов, так и из-за внешних обстоятельств. Потенциальные риски рейтинговой системы: [[8]](https://skillbox.ru/media/management/upravlenie-riskami-v-proekte-kak-nayti-i-otsenit-ikh-i-kak-sostavit-plan-zashchity/?ysclid=lpf8vkhgve599998703)

Таблица 1. Риски

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Риск** | **Вероятность** | **Описание** |
| 1 | Зависимость от качества данных | Средняя | Качество данных об оценках и производительности студентов может быть недостаточным или неточным. |
| 2 | Недобросовестное использование | Средняя | Риск злоупотребления системой для манипуляций данными или неправомерного влияния на процесс оценки. |
| 3 | Технические сбои | Средняя | Возможность технических сбоев или сбоев в работе, влияющих на доступность и надежность системы. |
| 4 | Недостаточное тестирование | Зависит от уровня тестирования | Риск обнаружения критических ошибок и недоработок после запуска из-за недостаточного тестирования. |
| 6 | Нарушение конфиденциальности данных | Средняя | Риск утечки конфиденциальных данных студентов из-за недостаточной защиты системы. |

**Меры по снижению рисков**

|  |  |
| --- | --- |
| **Риск** | **Меры по снижению рисков** |
| Зависимость от качества данных | Улучшение сбора данных для повышения точности. Регулярная проверка данных на ошибки и их фильтрация. Обучение персонала для предотвращения ошибок. |
| Недобросовестное использование | Установка четких правил и политик использования системы. Мониторинг и аудит активности в системе. Обучение пользователей этике использования. |
| Технические сбои | Регулярные обновления и обслуживание системы. Резервное копирование данных и системы. Тестирование на уязвимости и стабильность. |
| Недостаточное тестирование | Обширное тестирование перед внедрением системы. Разработка детальных тестовых сценариев и случаев использования. Пилотное внедрение на ограниченной аудитории. |
| Нарушение конфиденциальности данных | Усиление безопасности с использованием современных методов шифрования и контроля доступа. Управление доступом только для авторизованных пользователей. Обучение сотрудников правилам безопасности данных. |

Таблица 2. Меры по снижению рисков

## **8. Выводы**

Проект по созданию и внедрению рейтинговой системы представляет собой перспективное решение для колледжа, направленное на улучшение образовательного процесса, мотивации студентов и формирования конкурентоспособных выпускников.

# Требования

**Спецификация требований программного обеспечения** — [структурированный](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) набор требований/запросов к [программному обеспечению](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и его внешним интерфейсам. Предназначен для того, чтобы установить базу для соглашения между [заказчиком](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BA%D0%B0%D0%B7%D1%87%D0%B8%D0%BA) и [разработчиком](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82) (или подрядчиками) о том, как должен функционировать [программный продукт](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). [[1]](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)

Эта система направлена на оценку успеваемости студентов и мониторинг их достижений, способствуя улучшению образовательного процесса и поддерживая студентов в их учебных целях. Для стабильной работы системы и положительного опыта эксплуатирования необходимы следующие требования:

1. **Надежность и доступность данных:** Система должна поддерживать экспорт, импорт и хранение данных.
2. **Расчет рейтинга:** Система должна поддерживать различные категории рейтинга, такие как академические оценки, посещаемость, участие во внеучебных мероприятиях и другие параметры, которые считаются важными для колледжа.
3. **Индивидуализация и настраиваемость:** Должна быть реализована возможность настройки параметров категорий рейтинга в зависимости от целей и приоритетов колледжа.
4. **Удобный интерфейс:** Интуитивно понятный пользовательский интерфейс, который позволит студентам, преподавателям и модераторам легко взаимодействовать с системой.
5. **Мобильная доступность:** Доступ к системе через мобильные устройства для удобства использования в любое время и в любом месте.
6. **Отчеты и аналитика:** Должна присутствовать возможность создания отчетов и аналитики для администрации колледжа, чтобы оценивать успехи и идентифицировать области для улучшения.
7. **Интеграция:** Интеграция с сайтом колледжа.
8. **Обратная связь и поддержка:** Механизмы обратной связи и технической поддержки для пользователей системы.
9. **Уведомления и оповещения:** Возможность отправки уведомлений для студентов и преподавателей о важных событиях.
10. **Система поощрений и наград:** Система должна предоставлять рекомендации для поощрения для мотивации студентов к достижению высоких результатов.

# 2. Архитектура

Архитектура веб-приложений — это план одновременных взаимодействий между компонентами, базами данных, промежуточными системами, пользовательскими интерфейсами и серверами в приложении. [[2]](https://heaad.ru/blog/arkhitektura-prilozheniy/?ysclid=lpf8mm283b22738209)

Для оптимальной работы рейтинговой системы на фреймворке Django используется следующая архитектура.

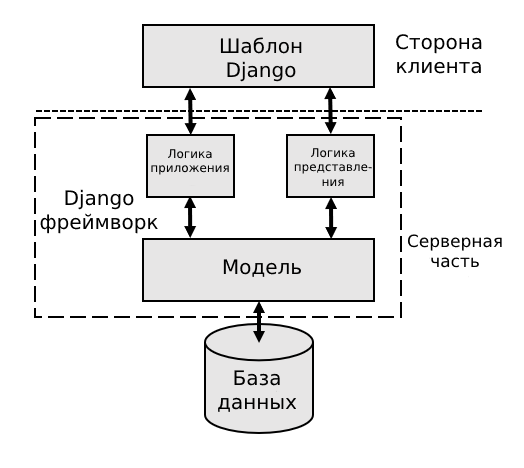


рис. 1. Архитектура проекта

# 4. Бизнес-процесс

Бизнес-процесс - это описание шагов и действий, которые сотрудники или участники организации выполняют для достижения определенной бизнес-цели или результата. [[3]](https://neiros.ru/blog/business/biznes-processy-kak-sposob-uluchshit-rabotu-predpriyatiya/?ysclid=lpf8p02u21985110945)

**IDEF0** — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временна́я последовательность (поток работ). [[4]](https://ru.wikipedia.org/wiki/IDEF0)

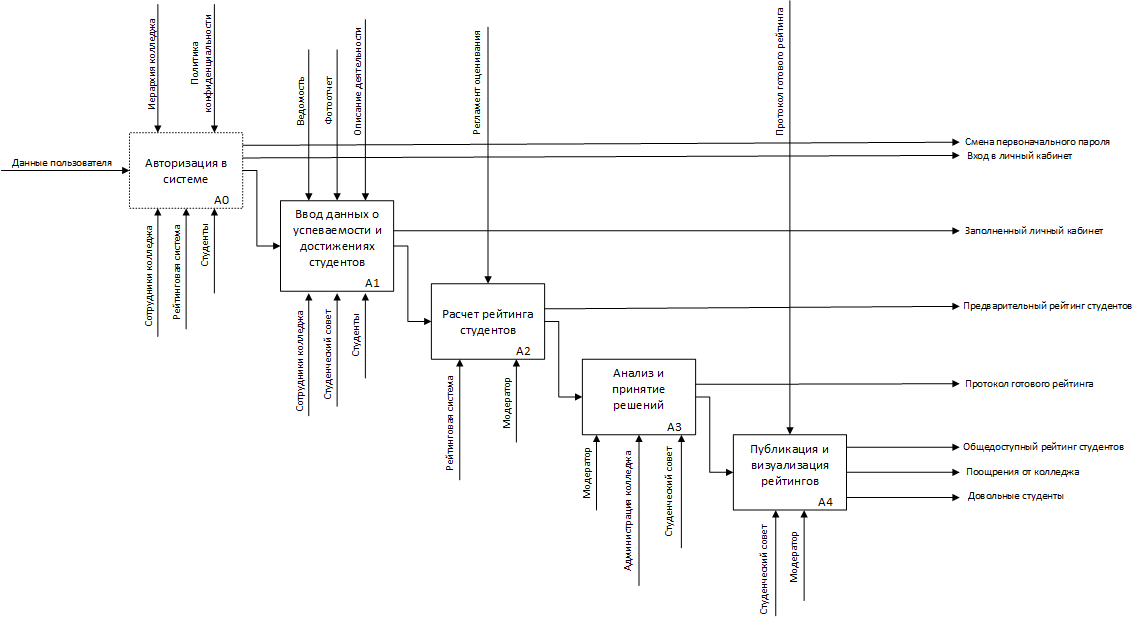


рис. 2. Модель IDEF0

# Заключение

В ходе данного исследования были выполнены задачи, направленные на создание инновационной и адаптивной рейтинговой системы для Новосибирского политехнического колледжа "TopNPK". Анализ существующих рейтинговых систем в образовании позволил выделить оптимальные практики и подходы к их проектированию. Определены критерии оценки и параметры, учитывающие особенности колледжа, что способствует справедливой оценке студенческих достижений.

На основе этих данных была разработана модель рейтинговой системы, предназначенной для "TopNPK". Эта система не только отвечает текущим потребностям образовательного процесса, но и обеспечивает более точную оценку учебных успехов студентов, способствуя их мотивации и активной деятельности.

В ходе исследования были определены эффективные методы сбора и анализа данных для работы рейтинговой системы. Проведенное тестирование на выборке студентов позволило подтвердить гипотезу о том, что внедрение инновационной рейтинговой системы в "TopNPK" действительно способствует более объективной оценке достижений студентов и повышению их мотивации.

Наконец, предложены рекомендации по внедрению и дальнейшему совершенствованию рейтинговой системы. Эти рекомендации ориентированы на улучшение ее функциональности и соответствия развивающимся требованиям образовательного процесса.

Таким образом, реализация предложенной рейтинговой системы в Новосибирском политехническом колледже "TopNPK" не только способствует повышению конкурентоспособности учебного заведения, но также обеспечивает более качественное образование, поддерживая студентов в их стремлении к успеху.

1 Bag Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id\_TC** | **Приложение** | **Подсистема** | **Действия** | **Ожидаемый Результат** | **Фактический Результат** | **Уровень ошибки** |

Сделать вывод, что ошибки были устранены и добавить модульные Тесты обязательно!

2 Меры информационной безопасности

3 Информационная безопасность (2 листа), ИСПОЛЬЗОВАТЬ МОДЕЛЬ ФСТЭК

Список угроз для системы, вставить таблицу.

|  |  |  |
| --- | --- | --- |
| Уязвимости | Угрозы | Меры для предотвращения угроз |
| Поль | 1. Намеренное повреждение данных  2. Программные сбои  3.Использования ПО несанкционированными пользователями  4. Незаконное использование По | 1. Установка пароля для доступа к данным  2. Тестирование программных модулей  3. Авторизация для доступа к данным |
| Среда и инфраструктура | Неисправность в системе электроснабжения | Установка бесперебойного питания |
| Программное обеспечение | Программные сбои | Тестирование программного обеспечения |
| Аппаратное обеспечение | Аппаратные отказы | Резервирование аппаратного обеспечения |

1. РЕЦЕНЗИЯ

* + - 1. <https://ppt-online.org/319777>
      2. <https://ru.wikipedia.org/wiki/Python>
      3. <https://practicum.yandex.ru/blog/chto-takoe-javascript-zachem-nuzhen/>
      4. <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction>
      5. <https://skillbox.ru/media/code/postgresql-vsye-chto-nuzhno-znat-dlya-bystrogo-starta/?ysclid=lsc2qnao5z769427253>
      6. <https://skillbox.ru/media/code/chto_takoe_html/?ysclid=lsc2ujne2w326993265>
      7. <https://practicum.yandex.ru/blog/chto-takoe-css/>
      8. <https://htmlacademy.ru/blog/git/git-console>
      9. <https://1cloud.ru/blog/django_one_server?ysclid=lsc36lbbiy352645385>
      10. <https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B3%D0%B5%D1%82>
      11. <https://habr.com/ru/articles/722054/>
      12. <https://www.yourtodo.ru/posts/celery/?ysclid=lsc3gu6swy430858743>
      13. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2>
      14. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2>

# Список литературы

1. <https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F>
2. <https://heaad.ru/blog/arkhitektura-prilozheniy/?ysclid=lpf8mm283b22738209>
3. <https://neiros.ru/blog/business/biznes-processy-kak-sposob-uluchshit-rabotu-predpriyatiya/?ysclid=lpf8p02u21985110945>
4. <https://ru.wikipedia.org/wiki/IDEF0>
5. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2>
6. <https://itonboard.ru/analysis/394-diagramma_posledovatelnosti_sequence_diagrams_uml/?ysclid=lpf8sunzrk739781306>
7. <https://www.lucidchart.com/pages/ru/erd-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0>
8. <https://skillbox.ru/media/management/upravlenie-riskami-v-proekte-kak-nayti-i-otsenit-ikh-i-kak-sostavit-plan-zashchity/?ysclid=lpf8vkhgve599998703>
9. <https://www.diasoft.ru/about/publications/20724/?ysclid=lpozu1183755174414>

от чьего лица повествовать?

В единственном или множественном числе?

Заменить постгри на скюллите

Какого шрифта должны быть таблицы

Заменить диаграмму прецедентов

Подписать рисунки/таблицы

Как оформить списки с отступами?