**План обеспечения качества**

**1. Цели**

* Убедитесь, что система точно регистрирует и обновляет академические рейтинги и «долги» студентов.
* Обеспечьте надежные функции поиска и фильтрации для диапазонов оценок.
* Обеспечить производительность, удобство использования и безопасность системы для всех пользователей (студентов, преподавателей, администраторов).

**2. Область применения**

* **Основные характеристики** :
  + Хранение и обновление рейтингов студентов по определенным дисциплинам.
  + Просмотр информации об успеваемости учащихся.
  + Поиск студентов по указанному диапазону рейтингов.
  + Регистрация и управление «долгами» (например, невыполненными предметами или неуспешно выполненными заданиями).
* **Исключено** : интеграция с внешними системами, если не указано иное.
* **Пользователи** : преподаватели, администраторы и студенты.

**4. Стандарты и процедуры**

* Соблюдайте стандарты **IEEE 730** для документации по контролю качества.
* Используйте рекомендации **ISO 25010 по качеству программного обеспечения:**
  + **Функциональность** : Точность сохраненных данных и результатов поиска.
  + **Удобство использования** : система должна быть интуитивно понятной и простой в использовании.
  + **Производительность** : Система должна обрабатывать большое количество записей без задержек.
  + **Безопасность** : защита конфиденциальных данных учащихся от несанкционированного доступа.

**5. Процедуры испытаний**

**а) Функциональное тестирование**

* **Рейтинги записи** :
  + Тестовое хранение и обновление рейтингов студентов по различным дисциплинам.
  + Убедитесь, что неверные или неполные записи отклоняются.
* **Просмотр информации** :
  + Протестируйте интерфейс для получения и отображения сведений об успеваемости учащихся.
  + Убедитесь, что данные отображаются правильно и полно.
* **Функциональность поиска** :
  + Протестируйте функцию поиска с различными диапазонами рейтинга.
  + Убедитесь, что в результатах отображаются только те студенты, которые соответствуют критериям.
* **Управление «долгами»** :
  + Попробуйте добавить, обновить и удалить «долги».
  + Убедитесь, что долги связаны с правильной записью студента.

**б) Тестирование удобства использования**

* Оцените интуитивность дизайна интерфейса.
* Проверьте удобство использования навигации (например, поиск функций поиска, просмотра или обновления).
* Собирайте отзывы от конечных пользователей (учителей и администраторов).

**в) Тестирование производительности**

* Тестирование поведения системы с использованием большого набора данных.
* Проверить время отклика системы (менее 3 секунд для операций поиска).
* Имитация одновременного доступа пользователей к системе.

**г) Тестирование совместимости**

* Тестируйте на нескольких устройствах (настольных компьютерах, ноутбуках, планшетах).
* Проверьте работоспособность в основных браузерах (Chrome, Firefox, Edge) и операционных системах (Windows, macOS).

**6. Показатели качества**

* **Функциональная точность** :
  + 100% введенных данных и обновлений должны храниться правильно.
* **Точность поиска** :
  + Результаты поиска должны соответствовать критериям с точностью 98% или выше.
* **Производительность** :
  + Система должна отвечать на запросы менее чем за 3 секунды.
* **Удобство использования** :
  + Не менее 90% конечных пользователей должны оценить систему как «простую в использовании».
* **Безопасность** :
  + В ходе тестирования критических уязвимостей не обнаружено.

**7. Управление рисками**

* **Потенциальные риски** :
  + Потеря или повреждение данных во время хранения или обновления.
  + Несанкционированный доступ к конфиденциальным данным студентов.
  + Низкая производительность при работе с большими наборами данных.
* **Стратегии смягчения последствий** :
  + Внедряйте надежное резервное копирование баз данных.
  + Внедряйте надежные протоколы аутентификации и шифрования.
  + Оптимизируйте запросы к базе данных и индексацию для поисковых операций.

**8. Постоянное совершенствование**

* Регулярно собирайте отзывы пользователей для улучшения системы.
* Ежеквартально просматривайте результаты тестов и показатели, чтобы выявить области для улучшения.
* Внедряйте новые технологии или инструменты для повышения надежности и безопасности системы.

**9. Документация и отчетность**

* **Тестовые случаи** : Поддерживайте подробные тестовые случаи для всех функций.
* **Отчеты о дефектах** : документируйте все дефекты с указанием уровней серьезности и шагов по их воспроизведению.
* **Отчеты о ходе работ** : делитесь еженедельными отчетами о ходе работ по обеспечению качества с заинтересованными сторонами.
* **Итоговый отчет по контролю качества** : перед развертыванием предоставьте подробный отчет по контролю качества.

**Спринты и задачи**

**Спринт 1: Настройка проекта (предварительная разработка)**

**Цель** : Подготовить основу для развития.

**Задачи:**

1. **Сбор требований** :
   * Встретьтесь с заинтересованными сторонами (например, учителями, администраторами) для окончательного согласования системных требований.
   * Создавайте пользовательские истории (например, «Как учитель, я хочу регистрировать оценки учеников, чтобы отслеживать их успеваемость»).
2. **Архитектура системы проектирования** :
   * Определите схему базы данных системы (например, таблицы для студентов, рейтингов, дисциплин и задолженностей).
   * Выберите технологический стек (например, Python для бэкэнда, Jinja для фронтэнда, SQLite для базы данных).
3. **Настройка среды разработки** :
   * Инструменты настройки ( Flask,SQLite ,).
   * Настройте серверы или облачную инфраструктуру (Git).

**Результаты** :

* Документированные требования и истории пользователей.
* Проектирование схемы и архитектуры базы данных.
* Полностью настроенная среда разработки.

**Спринт 2: Базовая системная инфраструктура**

**Цель** : Создание основополагающих компонентов системы.

**Задачи:**

1. **Разработка бэкэнда** :
   * Создание конечных точек для операций CRUD (создание/чтение/обновление/удаление) для студентов и дисциплин.
2. **Разработка интерфейса** :
   * Разработать базовые компоненты пользовательского интерфейса (панель управления).
   * Отобразить список студентов с указанием их дисциплин и рейтингов.
3. **Настройка базы данных** :
   * Создавайте и заполняйте таблицы по студентам, дисциплинам и рейтингам.

**Результаты** :

* Прототип панели приборов.
* Базовая функциональность CRUD для студентов и дисциплин.

**Спринт 3: Основные функции (рейтинги и долги)**

**Цель** : Разработать основные функции, такие как хранение рейтингов и управление задолженностями.

**Задачи:**

1. **Разработка бэкэнда** :
   * Добавить API для записи и обновления оценок студентов.
   * Добавить API для регистрации и обновления «долгов», связанных со студентами.
2. **Разработка интерфейса** :
   * Создайте формы для ввода оценок и «долгов» учителями.
   * Отображение сведений о студентах, рейтингов и «долгов» на панели управления.
3. **Тестирование** :
   * API-интерфейсы модульного тестирования для обеспечения точной обработки данных.
   * Проверка запросов к базе данных для хранения и извлечения оценок.

**Результаты** :

* Функциональная система рейтингования и управления долгом.
* Базовый пользовательский интерфейс для преподавателей и администраторов.

**Спринт 4: Функции поиска и фильтрации**

**Цель** : Реализовать возможности поиска и фильтрации.

**Задачи:**

1. **Разработка бэкэнда** :
   * Создавайте API для поиска студентов по идентификатору, имени или диапазону рейтингов.
   * Добавьте функции сортировки и разбиения на страницы для больших наборов данных.
2. **Разработка интерфейса** :
   * Разработайте панель поиска и параметры фильтра.
   * Отображение результатов поиска в режиме реального времени с разбивкой на страницы.
3. **Тестирование** :
   * Проверьте точность поиска с помощью различных диапазонов рейтинга и критериев.
   * Проверка производительности API при различных условиях нагрузки.

**Результаты** :

* Функция поиска и фильтрации на странице списка студентов.

**Спринт 5: Улучшения и проверка пользовательского интерфейса**

**Цель** : повысить удобство использования и обеспечить точность данных.

**Задачи:**

1. **Разработка интерфейса** :
   * Добавить проверку входных данных для рейтингов и долгов.
   * Реализуйте адаптивный дизайн для мобильных устройств и планшетов.
2. **Улучшения пользовательского опыта** :
   * Добавьте параметры сортировки (например, по идентификатору, имени, дисциплине или рейтингу).
   * Предоставляйте подробные подсказки и сообщения об ошибках для руководства пользователя.
3. **Тестирование** :
   * Проведите тестирование удобства использования с преподавателями и администраторами.
   * Проверка адаптивности на разных устройствах и в разных браузерах.

**Результаты** :

* Улучшенный, удобный интерфейс с проверкой.

**Спринт 6: Финальное тестирование и развертывание**

**Цель** : Завершить тестирование, устранить оставшиеся проблемы и развернуть систему.

**Задачи:**

1. **Тестирование системы** :
   * Проведите комплексное тестирование всех функций.
   * Проведите стресс-тестирование и нагрузочное тестирование для выявления узких мест.
2. **Обучение пользователей** :
   * Организуйте обучающие занятия или предоставляйте руководства для преподавателей и администраторов.
3. **Развертывание** :
   * Развертывание системы в производственной среде.
   * Контролируйте первоначальное использование системы и сообщения об ошибках.

**Результаты** :

* Полностью протестированная и развернутая система.

**Хронология спринта (приблизительная)**

|  |  |  |
| --- | --- | --- |
| **Спринт** | **Продолжительность** | **Ключевая веха** |
| Спринт 1 | 2 недели | Настройка и архитектура проекта завершены |
| Спринт 2 | 2 недели | Базовая инфраструктура и функциональность CRUD |
| Спринт 3 | 2 недели | Основная система рейтинга и управления долгом |
| Спринт 4 | 2 недели | Функции поиска и фильтрации |
| Спринт 5 | 2 недели | Улучшения пользовательского интерфейса |
| Спринт 6 | 2 недели | Окончательное тестирование и развертывание |