Техническое задание (ТЗ) на разработку программы для записи книг и сериалов

**1. Введение**

Разрабатываемая программа “StoryTracker” предназначена для создания и управления списком книг и сериалов, которые пользователь хочет прочитать или посмотреть. Программа должна предоставлять удобный интерфейс для добавления, удаления, отметки статуса прочтения/просмотра и отображения записей о книгах и сериалах. Программа должна обеспечивать сохранение данных в базе данных для сохранения информации между сессиями.

**2. Цели и задачи**

* Создать программу для ведения личного списка книг и сериалов.
* Предоставить пользователю возможность добавлять записи о книгах и сериалах, включая название, автора (для книг) или создателей (для сериалов).
* Разработать механизм для маркировки книг/сериалов как прочитанных/просмотренных.
* Обеспечить возможность удаления записей о книгах/сериалах.
* Реализовать удобный и наглядный интерфейс для взаимодействия с программой.
* Обеспечить сохранение данных в базе данных PostgreSQL.
* Разработать систему контроля версий с использованием Git.
* Провести тестирование программы с использованием PyTest.

**3. Функциональные требования**

**3.1 Основные функции**

1. **Добавление книги/сериала:**
   * Пользователь должен иметь возможность ввода названия книги/сериала.
   * Для сериалов можно ввести информацию о создателях (например, название, имена создателей).
   * Программа должна валидировать вводимые данные. Необходимо предоставить сообщение об ошибке, если введены некорректные данные, например, пустое поле названия книги.
   * Данные о книге/сериале должны сохраняться в базе данных.
2. **Отметка о прочтении/просмотре:**
   * Пользователь должен иметь возможность отметить книгу/сериал как прочитанный/просмотренный.
   * Изменение статуса должно отражаться в базе данных и в отображении списка.
   * Необходимо предоставить опцию выбора между “Прочитано”, “Не прочитано”, “Просмотрено”, “Не просмотрено” для книг и сериалов соответственно.
3. **Удаление записи:**
   * Пользователь должен иметь возможность удалять записи из списка.
   * При удалении записи данные должны быть удалены из базы данных.
4. **Отображение списка:**
   * Программа должна отображать список книг и сериалов в удобном формате.
   * Список должен включать название, статус (прочитано/не прочитано, просмотрено/не просмотрено).
5. **Сохранение и загрузка данных:**
   * Все данные должны сохраняться в базе данных PostgreSQL.
   * Программа должна загружать данные из базы данных при запуске.
   * Программа должна обеспечивать резервное копирование данных в базу данных.

**3.2 Параметры пользовательского интерфейса**

* Интерфейс должен быть интуитивно понятным и удобным в использовании.
* Использовать модуль PyQt5 для создания визуально привлекательного интерфейса.
* Меню должно быть навигационным и разграниченным по группам функций.
* Должны быть визуальные индикаторы (Цвет текста) для отображения текущего статуса книги/сериала (прочитано/не прочитано, просмотрено/не просмотрено).

**4. Нефункциональные требования**

* Язык программирования: Python 3.x.
* Операционная система: Windows.
* Программа должна работать без ошибок.
* Код должен быть хорошо документирован.
* Программа должна быть написана с использованием принципов ООП.
* Программа должна обеспечивать стабильную работу при работе с большим количеством записей.

**5. Технологии и инструменты**

* Язык программирования: Python
* База данных: PostgreSQL.
* Структуры данных: списки, словари, классы Python (для более эффективной организации данных).
* Модуль: PyQt5
* Система контроля версий: Git
* Тестирование: PyTest

**6. Этапы разработки**

1. **Подключение базы данных:** Настройка соединения с PostgreSQL, создание таблиц для книг и сериалов.
2. **Реализация основных функций:** Разработка функций для добавления, удаления, изменения статуса и отображения списка. Проверка валидности входных данных.
3. **Разработка интерфейса:** Создание графического интерфейса с использованием PyQt5.
4. **Тестирование:** Полное тестирование программы с помощью PyTest, включая тестирование на ошибки и граничные случаи.
5. **Документирование:** Добавление документации и комментариев к коду, руководство пользователя для интерфейса.
6. **Релиз:** Размещение программы в Git repository и сборка/дистрибуция.

**7. Критерии приемки**

* Программа должна корректно запускаться и завершаться.
* Все функции должны работать согласно функциональным требованиям.
* Данные должны сохраняться и загружаться корректно.
* Пользовательский интерфейс должен быть интуитивно понятным и удобным.
* Программа должна быть протестирована с использованием PyTest и не иметь критических ошибок.
* Программа должна быть документирована.

**8. Временные рамки**

* Оценка сроков выполнения проекта: 2 недели.