**Руководство программиста**

**«**Информационной системы для магазина настольных игр»

1. **Инсталляция программы:**
   1. **Пользовательский вариант**
      1. Перейти на сайт источника(официальный сайт разработчика или репозитории на GitHub)
      2. Скачать файл формата \*.exe
      3. Проверить приложение.
   2. **Вариант поддержания и модернизации**
2. Убедитесь, что на вашем компьютере установлен Python, а также установлена программа для работы с бд – SQliteStudio.
3. Склонируйте репозиторий программы с GitHub с помощью команды git clone.
4. Перейдите в папку вашего проекта с помощью команды cd repository.
5. Установите зависимости проекта, выполнив команду pip install -r requirements.txt.
6. Запустите программу для проверки успешности установки командой – Python название\_программы.py.
7. Запустите программу нажав Enter.
8. **Проверка работоспособности ИС**

Для проверки работоспособности требуется запустить программу, проверить отображение таблиц и корректную работу кнопок приложения.

1. **Поддержка программы**

Регулярно обновляйте Python и PyQt5 до актуальных версий, чтобы поддерживать совместимость и использовать все последние улучшения производительности и безопасности.

Постоянно отслеживайте сообщения об ошибках от пользователей, анализируйте их, предоставляйте оперативную поддержку и своевременно исправляйте выявленные проблемы, чтобы минимизировать неудобства. При добавлении новых функций или внесении изменений уведомляйте пользователей, публикуя информацию на GitHub или других используемых платформах, чтобы обеспечить их осведомленность о новых возможностях программы.

Обновляйте документацию программы, включая инструкции по установке и устранению неполадок, раздел с часто задаваемыми вопросами (FAQ) и другую полезную информацию.

Храните актуальную версию документации в репозитории для удобного доступа. В целях обеспечения производительности регулярно оптимизируйте код и базу данных, чтобы программа работала максимально эффективно.

Также не забывайте о резервном копировании данных: регулярное создание резервных копий важно для предотвращения потери данных в случае непредвиденных обстоятельств.

1. **Модернизация**

Постоянно собирайте обратную связь, анализируя отзывы пользователей, чтобы выявлять возможности для улучшения программы и делать её более удобной и функциональной.

Для разработки новых функций создайте отдельную ветку командой git checkout -b feature-branch, что позволит вести работу над изменениями изолированно.

Реализуйте новые функции, используя Python и PyQt5 для создания логики и интерфейса программы.

Перед выпуском новых возможностей обязательно проводите тестирование для выявления возможных ошибок и повышения стабильности работы.

После завершения разработки обновите документацию, включив описание новых функций и изменений, чтобы пользователи могли легко понять, как использовать новые возможности.

Затем отправьте изменения в основной репозиторий, используя команду git push origin feature-branch, чтобы сохранить работу и сделать её доступной для команды или других пользователей.

Для поддержания программы в актуальном состоянии регулярно обновляйте зависимости с помощью команды pip install -r requirements.txt, что гарантирует совместимость и стабильность работы программы.