1. Общие сведения о программе

1.1 Название программы:
Прототип приложения "Агрегатор по поиску автозапчастей" — это программное обеспечение, предназначенное для поиска, сравнения и выбора автозапчастей на основе введенных пользователем параметров автомобиля. Программа интегрируется с внешними источниками данных (интернет-магазины) для получения актуальной информации о наличии, стоимости и характеристиках запчастей.

1.2 Назначение программы:
Программа предназначена для облегчения поиска автозапчастей через интернет, предоставляет пользователям возможность фильтровать и сортировать результаты, а также производить сравнение запчастей по нескольким критериям.

1.3 Функциональные возможности:

- Поиск запчастей по параметрам автомобиля.
- Сравнение запчастей по цене, производителю, наличию.
- Фильтрация и сортировка результатов поиска.
- Подключение к интернет-магазинам для получения актуальных данных.
- Добавление запчастей в список, избранных для последующего просмотра.
- Переход к продавцам через сайты.

1.4 Основные ограничения:

- Программа требует наличия постоянного интернет-соединения для работы с торговыми площадками.
- Программа может работать на Android и IOS.
- Требования к аппаратным ресурсам: процессор 2.0 ГГц, оперативная память 4 ГБ, свободное место на диске 500 МБ.

---

2. Структура программы

Программа состоит из нескольких основных компонентов, каждый из которых выполняет конкретные функции:

2.1. Интерфейс пользователя (UI):
Отвечает за взаимодействие с пользователем. Включает формы ввода параметров поиска, отображение результатов, фильтрацию и сортировку данных.

2.2. Модуль поиска и парсинга данных:
Получает запросы от интерфейса пользователя и выполняет запросы к внешним API интернет-магазинов. Этот модуль парсит и обрабатывает полученные данные, фильтруя их по заданным критериям.

2.3. Модуль сравнения запчастей:
После получения данных, данный модуль позволяет пользователю сравнивать запчасти по характеристикам (цене, производителю, наличию и т.д.).

2.4. Модуль работы с базой данных:
Сохраняет информацию о пользователях, истории их запросов, избранных запчастях. Также этот модуль отвечает за хранение и извлечение данных о предыдущих поисках.

2.5. Системы хранения данных:
База данных для хранения информации о пользователях и их избранных запчастях. Используется SQL-запросы для обработки данных.

---

3. Настройка программы

3.1. Требования к окружению: для правильной работы программы требуется следующее окружение:

- Операционная система: Android и IOS.
- Установленные следующие компоненты:
  - Python 3.x
  - Node.js (для клиента)
  - База данных PostgreSQL или SQLite
  - Пакеты Python: Requests, BeautifulSoup, Flask/Django
  - Библиотеки для работы с фронтендом: React.js, Redux (если используете state management)

3.2. Установка программы:

1. Скачайте архив с исходным кодом программы.
2. Установите необходимые библиотеки с помощью `pip install -r requirements.txt`.
3. Установите зависимости для клиента (если используется React.js или другая фронтенд-технология).
4. Подключите внешние API магазинов через их ключи доступа.
5. Настройте файл конфигурации для работы с базой данных (например, `settings.py` для Django или `config.py` для Flask).
6. Запустите программу с помощью команды `python manage.py runserver` (для Django) или соответствующей команды для выбранного фреймворка.

3.3. Конфигурация базы данных:

1. Для PostgreSQL: создайте базу данных и пользователя.
2. В файле конфигурации укажите параметры подключения к базе данных.
3. Примените миграции базы данных с помощью команд:
   - `python manage.py migrate` (для Django).
4. Для SQLite просто укажите путь к базе данных.

---

4. Проверка программы

4.1. Подготовка к тестированию: перед тестированием необходимо убедиться в правильной настройке всех компонентов:

- Настройте подключение к базе данных и внешним API.
- Убедитесь, что все зависимости установлены корректно.
- Проверьте, что интерфейс пользователя работает без ошибок.

4.2. Виды тестирования:

1. **Тестирование функциональности:**
   Проверка, что все ключевые функции программы (поиск запчастей, сравнение, фильтрация) работают корректно.
2. **Тестирование интеграции с API:**
   Проверьте, что программа корректно подключается к внешним API интернет-магазинов и корректно обрабатывает данные (наличие запчастей, цены, описание).
3. **Тестирование интерфейса пользователя:**
   Проверка работы пользовательского интерфейса: все формы, кнопки, элементы управления должны быть доступны и работать корректно.
4. **Тестирование производительности:**
   Проверка скорости работы программы при большом объеме данных. Тестирование работы при параллельных запросах к внешним API.
5. **Тестирование базы данных:**
   Проверьте правильность работы с базой данных, включая сохранение и извлечение информации, а также обработку ошибок при отсутствии соединения с базой.

4.3. Ошибки и их устранение:

1. **Проблемы с подключением к API:**
   Убедитесь, что ключи API верны, а также проверьте настройки подключения (например, тайм-ауты и ограничения по количеству запросов).
2. **Проблемы с базой данных:**
   Проверьте настройки подключения, корректность миграций и выполнение SQL-запросов.
3. **Проблемы с интерфейсом:**
   Проверьте консольные сообщения об ошибках в браузере или на сервере для выявления проблем с фронтендом или бэкендом.

---

5. Дополнительные возможности

5.1. Возможности расширения функционала:

- **Поддержка дополнительных API:** Возможность интеграции с новыми интернет-магазинами и API для расширения базы данных.
- **Рекомендательные системы:** Внедрение алгоритмов машинного обучения для предложения пользователям запчастей, подходящих под их предпочтения.
- **Мобильная версия:** Разработка мобильной версии приложения для пользователей смартфонов.

5.2. Дополнительные модули:

- **Модуль отзывов:** Возможность добавления и отображения отзывов пользователей о запчастях и магазинах.

- **Модуль уведомлений:** Отправка уведомлений пользователю о скидках или изменениях цен на избранные запчасти.

---

6. Сообщения системному программисту

6.1. Ошибки в программе:

- **Ошибка подключения к API:** "Не удается подключиться к серверу API. Проверьте настройки соединения."
- **Ошибка при работе с базой данных:** "Ошибка при выполнении запроса к базе данных. Проверьте структуру данных."
- **Ошибка пользовательского интерфейса:** "Ошибка валидации данных на форме поиска. Проверьте корректность введенных данных."

6.2. Ожидаемые действия:

- Для исправления ошибок необходимо сначала проверить логи программы, а затем устранить проблему на уровне кода или конфигурации.
- Если ошибка связана с внешними сервисами (API, база данных), необходимо связаться с их технической поддержкой.