### Requirement analysis: APP\_1 - Quản lý thông tin khách hàng

**Nhiệm vụ chính của APP\_1**:

1. **Nhập hoặc đọc thông tin khách hàng**:
   * **Thông tin cần lưu trữ**: ID khách hàng, tên, số điện thoại, loại xe, biển số xe, trạng thái sửa chữa (chưa sửa, đang sửa, đã sửa xong).
   * **Nguồn nhập liệu**: Có thể nhập từ console (CLI) hoặc từ các file dữ liệu (Excel, JSON). Nếu dữ liệu nhập từ file, APP\_1 sẽ cần có khả năng đọc và xử lý dữ liệu từ các định dạng này.
     + **Đọc từ JSON**: Dữ liệu có thể là các đối tượng JSON với các trường như id, name, phone, car\_type, license\_plate, repair\_status.
2. **Lưu trữ và cập nhật dữ liệu**:
   * Dữ liệu khách hàng cần được lưu trữ trong một file JSON hoặc Excel, tùy theo yêu cầu.
   * APP\_1 phải có khả năng **cập nhật**, **xóa** hoặc **thêm mới** khách hàng trong hệ thống dữ liệu.
3. **IPC (Inter-Process Communication)**:
   * APP\_1 cần gửi các thay đổi (thêm mới, cập nhật khách hàng) sang APP\_2 thông qua IPC (có thể sử dụng **sockets**, **message queues**, hoặc **shared memory**). Điều này giúp cho APP\_2 có thể xử lý thông báo và cập nhật thông tin trong thời gian thực.
4. **Quản lý thông tin dữ liệu**:
   * APP\_1 sẽ cần quản lý và duy trì danh sách khách hàng hiện có trong bộ nhớ. Khi có thao tác nhập dữ liệu từ file, danh sách khách hàng phải được cập nhật ngay lập tức.
   * Cung cấp giao diện cho phép **thêm** hoặc **xóa** khách hàng, đảm bảo rằng những thay đổi này được lưu lại vào file.

**APP\_2 - Hiển thị thông báo và tìm kiếm thông tin khách hàng**

**Nhiệm vụ chính của APP\_2**:

1. **Nhận thông báo khách hàng mới từ APP\_1**:
   * APP\_2 sẽ nhận thông báo từ APP\_1 khi có khách hàng mới được thêm vào hệ thống hoặc khi có thay đổi về trạng thái sửa chữa của khách hàng.
   * Thông báo này có thể là một **alert** hoặc **notification** hiển thị trên giao diện của APP\_2, giúp người dùng biết rằng có một khách hàng mới hoặc có sự thay đổi cần chú ý.
2. **Tìm kiếm khách hàng**:
   * APP\_2 cần cung cấp tính năng tìm kiếm khách hàng theo **ID** hoặc **tên**. Điều này có thể được thực hiện thông qua một giao diện dòng lệnh hoặc thông qua các API.
   * Tính năng tìm kiếm cần được tối ưu sao cho khi nhận yêu cầu tìm kiếm, APP\_2 có thể trả về thông tin khách hàng nhanh chóng và chính xác.
3. **Cập nhật hóa đơn và lịch sử sửa chữa**:
   * APP\_2 cần cung cấp giao diện để **cập nhật hóa đơn sửa chữa** và **lịch sử sửa chữa** cho mỗi khách hàng. Điều này có thể bao gồm các thông tin như ngày sửa chữa, loại sửa chữa, giá cả, và các ghi chú quan trọng khác.
   * Lịch sử sửa chữa có thể được lưu trữ trong cơ sở dữ liệu (hoặc file) và được liên kết với thông tin khách hàng.
4. **Giao tiếp với APP\_1 qua IPC**:
   * APP\_2 sẽ cần giao tiếp với APP\_1 để nhận thông tin khách hàng hoặc cập nhật thông tin khách hàng mới. Đây là một bước quan trọng để cả hai ứng dụng có thể hoạt động đồng bộ.
   * Các thông tin có thể được truyền qua **sockets** (TCP/IP hoặc UDP).
5. **Xử lý và hiển thị dữ liệu**:
   * APP\_2 có thể là một **CLI** hoặc **GUI** ứng dụng (CLI theo yêu cầu), nơi thông tin khách hàng được hiển thị dưới dạng bảng hoặc danh sách. Việc này giúp người dùng dễ dàng theo dõi tình trạng sửa chữa của xe hoặc tìm kiếm thông tin khách hàng.

**Giao tiếp giữa APP\_1 và APP\_2**

**IPC** (Inter-Process Communication) sẽ là phương thức chính để truyền tải dữ liệu giữa hai ứng dụng:

* **Sockets**: Đây là một trong những cách phổ biến để giao tiếp giữa các ứng dụng trong môi trường C++. Các thông tin như thông báo khách hàng mới, cập nhật trạng thái, hoặc yêu cầu tìm kiếm có thể được gửi qua một socket TCP/IP hoặc UDP.

**Kết luận**

* **APP\_1** sẽ đảm nhận phần nhập liệu và lưu trữ thông tin khách hàng, trong khi **APP\_2** sẽ xử lý thông báo, tìm kiếm, cập nhật hóa đơn, và theo dõi sửa chữa. Cả hai ứng dụng sẽ giao tiếp qua IPC để đồng bộ hóa dữ liệu.