# Lời mở đầu

Trong bối cảnh cách mạng công nghiệp 4.0 và kinh tế nước nhà đang phát tiển như hiện nay, việc ứng dụng công nghệ thông tin vào các ngành nghề truyền thống đang ngày càng trở nên cần thiết để bắt kịp xu hướng phát triển đất nước cũng như phát triển kinh tế địa phương. Sản xuất rượu các sản phẩm thủ công truyền thống hay rượu làng nghề không chỉ là một nét văn hóa đặc trưng mà còn là một ngành nghề mang lại thu nhập cho người dân. Tại các vùng quê thuộc huyện Kim Sơn, tỉnh Ninh Bình công việc sản xuất sản phẩm rượu gạo này vẫn được duy trì, phát triển và là kế sinh nhai của nhiều hộ gia đình. Tuy nhiên, quá trình sản xuất và kinh doanh rượu tại các hộ gia đình đơn lẻ này còn đang sơ sài, hiện nay vẫn còn gặp nhiều khó khăn trong việc quản lý nguyên liệu, theo dõi quy trình sản xuất, thống kê tài chính, và kiểm soát kho hàng. Những vấn đề này gây ra không ít thách thức trong việc nâng cao hiệu quả hoạt động, duy trì chất lượng sản phẩm và giảm khả năng cạnh tranh trên thị trường. Việc áp dụng công nghệ thông tin vào quản lý sản xuất rượu giúp hiện đại hóa quy trình sản xuất, tăng hiệu quả cạnh tranh trên thị trường mà vẫn giữ lại được nét mộc mạc truyền thống của quê hương.

Đề tài "Xây dựng phần mềm quản lý sản xuất rượu RuKiSo" ra đời với mục tiêu giải quyết những khó khan kể trên. Phần mềm này được thiết kế nhằm hỗ trợ việc quản lý nhập xuất các sản phẩm cũng như nguyên liệu đầu vào, theo dõi các mẻ ủ rượu và thống kê tài chính chi tiết một cách có hệ thống. Với những tính năng này, RuKiSo không chỉ giúp tăng cường hiệu quả quản lý mà còn đóng góp vào việc chuẩn hóa và hiện đại hóa quy trình sản xuất rượu thủ công, rượu làng nghề.

Đối tượng nghiên cứu của đề tài là quy tình sản xuất và kinh doanh tại các hộ dân sản xuất rượu tại huyện Kim Sơn, tỉnh Ninh Bình với quy mô nhỏ lẻ và quy trình lạc hậu. Phạm vi nghiên cứu của đề tài bao gồm các nghiệp vụ quản lý từ các nguyên liệu đầu vào, quy trình sản xuất cho tới sản phẩm đầu ra và thống kê tài chính.

Với việc ứng dụng các công nghệ hiện đại như .NET API, SQL Server và .NET MAUI, phần mềm không chỉ mang lại hiệu quả quản lý cao mà còn dễ sử dụng, phù hợp với người dân địa phương và dễ dàng nâng cấp để chạy trên mọi nền tảng sau này.

Đề tài kỳ vọng góp phần hiện đại hóa quy trình sản xuất rượu truyền thống, nâng cao giá trị kinh tế và bảo tồn nét đẹp văn hóa của nghề làm rượu Kim Sơn. Kết quả đạt được bao gồm một phần mềm quản lý sản xuất rượu hoạt động ổn định, giao diện thân thiện, dễ sử dụng với người Việt Nam và tích hợp các tính năng cần thiết để hỗ trợ cho các cơ sở sản xuất. RuKiSo kỳ vọng trở thành một công cụ hữu ích, góp phần nâng cao hiệu quả sản xuất kinh doanh và bảo tồn giá trị văn hóa của sản phẩm rượu Kim Sơn.

# Chương 1: Mô tả bài toán

## Mục tiêu của bài toán

Mục tiêu chính của đề tài “Xây dựng phần mềm quản lý sản xuẩ rượu RuKiSo” là tạo ra một phần mềm giúp quản lý và tối ưu hóa quy trình sản xuất rượu thủ công, rượu làng nghề tại các hộ gia đình ở huyện Kim Sơn, tỉnh Ninh Bình.

Đối tượng người dùng được RuKiSo hướng tới là những người không quen sử dụng công nghê nên muc tiêu của phần mềm còn là cung cấp giao diện trực quan sinh động, dễ sử dụng và thân thiện. Phần mềm này giúp cho người dùng quản lý tốt hơn các nguyên liệu đầu vào cũng như sản phẩm đầu ra, theo dõi và nhắc nhở quá trình lên men, tạo ra hệ thống báo cáo, thống kê tài chính góp phần nâng cao hiệu quả sản xuất và kinh doanh, giúp đỡ người dân trong phát triển kinh tế vào bảo tồn truyền thống quy báu.

## Các yêu cầu đặt ra của bài toán

### Yêu cầu về dữ liệu

Để đáp ứng được mục tiêu quản lý toàn diện quy trình sản xuất và kinh doanh, phần mềm RuKiSo cần xử lý và quản lý một lượng dữ liệu không quá phức tạp nhưng mang tính hệ thống và rõ ràng. Dữ liệu này đóng vai trò cốt lõi trong việc hỗ trợ người dùng thực hiện các nghiệp vụ quản lý hàng ngày một cách hiệu quả và chính xác. Các loại dữ liệu cụ thể mà phần mềm cần xử lý bao gồm:

* **Quản lý danh mục nguyên liệu đầu vào**: Tên nguyên liệu, đơn vị tính, ngày nhập
* Quản lý danh mục sản phẩm: Tên sản phẩm, giá bán, số lượng
* **Ghi nhận các thông tin tài chính**: Số vốn, doanh thu từ bán hàng
* **Quản lý quy trình sản xuất**: Ngày bắt đầu ủ, ngày sản xuất, sản lượng
* **Thống kê kinh doanh:** Số lượng và giá trị của sản phẩm bán được hoặc các nguyên liệu đầu vào

### Yêu cầu về chức năng xử lý

Mục tiêu của RuKiSo là hiện đại hóa và tối ưu hóa quy trình sản xuất rượu, phần mềm RuKiSo cần cung cấp các chức năng xử lý không chỉ đáp ứng được nhu cầu quản lý cơ bản mà còn hỗ trợ người dùng cải thiện hiệu suất công việc. Các chức năng này phải được thiết kế một cách khoa học, tập trung vào sự đơn giản, dễ sử dụng và khả năng mang lại giá trị thực tế cao cho người dùng. Dưới đây là các yêu cầu chức năng chính mà phần mềm cần phải đáp ứng:

* Thống kê các sản phẩm bán chạy, nguyên liệu được sử dụng nhiều
* Quản lý nguyên liệu: Thêm, sửa, xóa nguyên liệu và theo dõi tồn kho
* Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm và theo dõi tồn kho
* Quản lý quy trình sản xuất: Tạo vào theo dõi lịch sản xuất, nhắc nhờ ghi đến thời gian hoàn thành
* Thống kê tài chính: hiện thị doanh thu, chi phí theo tuần, tháng
* Tương tác người dùng: Giao diện thân thiện dễ sử dụng, biểu đồ trực quan

### Yêu cầu về quy trình xử lý

Một phần mềm quản lý hiệu quả không chỉ dừng lại ở việc cung cấp các chức năng cần thiết mà còn phải đảm bảo quy trình xử lý thông tin diễn ra mượt mà, chính xác và thuận tiện. Quy trình xử lý đóng vai trò quan trọng trong việc quyết định trải nghiệm người dùng, độ chính xác của dữ liệu và khả năng duy trì hoạt động ổn định của phần mềm. Vì vậy, RuKiSo cần thiết kế quy trình xử lý rõ ràng, chặt chẽ và thân thiện với người dùng, từ khâu nhập liệu đến khâu xử lý và xuất thông tin, nhằm đạt được hiệu quả tối ưu trong việc vận hành hệ thống. Các yêu cầu bao gồm:

* Quy trình nhập liệu đơn giản, dễ sử dụng và có thông báo nếu người dùng nhập chưa đúng
* Phần mềm phải chính xác và hoạt động ổn định

## Mô tả các nguồn tài liệu, dữ liệu, biểu mẫu liên quan

### Nguồn tài liệu

Các nguồn tài liệu phục vụ cho việc nghiên cứu và xây dựng phần mềm quản lý sản xuất rượu RuKiSo được thu thập từ thực tế sản xuất và kinh nghiệm trong ngành sản xuất rượu truyền thống của chính gia đình, ngoài ra các ý kiến đóng góp và kinh nghiệm tham khảo cũng như học hỏi từ các hộ - cơ sở sản xuất khác cũng là một nguồn tài liệu quý giá để phát triển phần mềm này:

* **Kinh nghiệm gia đình**: Gia đình em có nhiều năm kinh nghiệm và nhiều thế hệ hoạt động trong lĩnh việc sản xuất rượu truyền thống. Với việc sản xuất để tự sử dụng trong gia đình trong thời gian đầu sau đó là phát triển trở thành công việc kinh doanh đem lại thu nhập cũng như bảo tồn nét văn hóa truyền thống của làng nghề rượu Kim Sơn. Quá trình này mất nhiều năm nhưng cũng đem lại các kiến thức sâu sắc về các bước sản xuất, các nguyên liệu và quản lý một cách thủ công.
* **Học hỏi từ các hộ sản xuất khác**: Trong suốt quá trình hình thành và phát triển thương hiệu rượu Kim Sơn, việc học hỏi trao đổi và khảo sát giữa các hộ sản xuất là một hoạt động cần thiết để ngành nghề này được duy trì và phát triển. Các kiến thức và khó khan gặp phải cũng là nhưng nguồn tài liệu vô cùng quý giá để định hướng và phát triển phẩn mềm
* **Tài liệu hướng dẫn phát triển phần mềm**: Các tài liệu về lập trình về .NET MAUI .NET API và SQL Server cùng các kiến thức lập trình được giảng dạy trên trường giúp xây dựng và triển khai một cách nhanh chóng và hiệu quả

### Dữ liệu thực nghiệm

Dữ liệu thực nghiệm đóng vai trò quan trọng trong việc đảm bảo tính chính xác và phù hợp của phần mềm quản lý sản xuất rượu RuKiSo. Các dữ liệu này được thu thập từ thực tế sản xuất của gia đình và từ các hộ sản xuất khác trong khu vực, kết hợp với những bộ dữ liệu giả lập nhằm mô phỏng và kiểm tra các chức năng của phần mềm. Việc sử dụng dữ liệu thực nghiệm không chỉ phản ánh đúng thực trạng mà còn giúp phần mềm đáp ứng tốt nhu cầu quản lý thực tế của người dùng. Dưới đây là các nguồn dữ liệu cụ thể được sử dụng trong dự án.

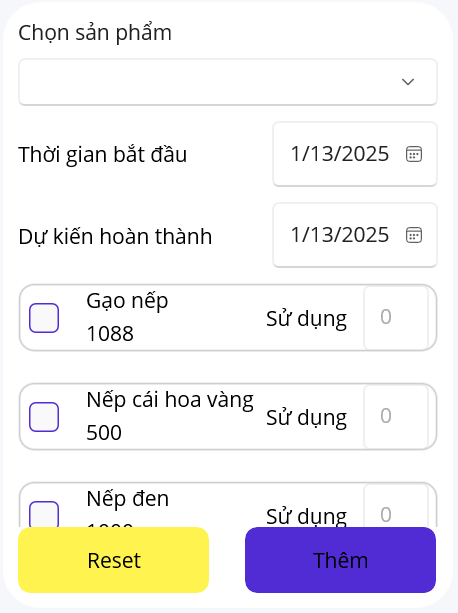
* **Dữ liệu từ sản xuất thực tế của gia đình**: Các ghi chép bao gồm các loại nguyên liệu (gạo nếp, men rượu), các giai đoạn sản xuất, sản lượng cũng như giá cả của các sản phẩm trong kho hang.
* **Dữ liệu tham khảo từ các hộ kinh doanh khác**: Cách họ quản lý, theo dõi quy trình sản xuất và tính toán chi phí trong kinh doanh
* **Dữ liệu giả lập để kiểm tra phần mềm**: Tạo ra bộ dữ liệu giả lập dựa trên kinh nghiệm thực tiển để kiểm tra các chức năng của phần mềm

### 1.3.3 Mẫu biểu

Mẫu biểu là yếu tố không thể thiếu để hỗ trợ việc thu thập, tổ chức và trình bày dữ liệu một cách trực quan, khoa học trong phần mềm quản lý sản xuất rượu RuKiSo. Các mẫu biểu này được thiết kế dựa trên thực tế hoạt động sản xuất và kinh doanh tại các hộ gia đình, nhằm đảm bảo tính tiện dụng và dễ hiểu cho người dùng. Các biểu mẫu cũng là cơ sở để phần mềm hiển thị thông tin và hỗ trợ người dùng trong việc theo dõi quy trình sản xuất, quản lý tồn kho và thống kê tài chính.

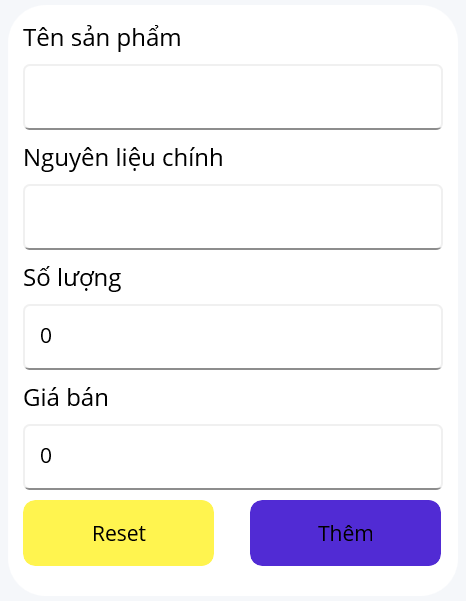
Dưới đây là một số loại mẫu biểu tiêu biểu được sử dụng trong dự án:

* Mẫu biểu tạo một mẻ rượu để theo dõi quy trình sản xuất. Mẫu biểu gồm có loại rượu thành phẩm, ngày tháng bắt đầu và dự kiến kết thúc ,các nguyên liệu có trong kho, được sử dụng bằng cách chọn và nhập số lượng



Hình 1-1 Mẫu biểu tạo mới mẻ rượu

* Mẫu biểu thêm mới sản phẩm bao gồm tên sản phẩm, các nguyên liệu chính, số lượng và giá bán



Hình 1-2 Mẫu biểu tạo mới sản phẩm

## Phân tích, đánh giá hiện trạng và những vấn đề còn tồn tại

### Hiện trạng sản xuất và kinh doanh rượu tại địa phương

Tại huyện Kim Sơn, tỉnh Ninh Bình, các hộ gia đình sản xuất rượu gạo truyền thống vẫn đang duy trì quy trình sản xuất lâu đời, mang đậm nét văn hóa làng nghề. Tuy nhiên, hoạt động sản xuất và kinh doanh chủ yếu diễn ra theo cách thủ công với các đặc điểm:

* Chủ yếu sử dụng sổ sách giấy hoặc trí nhớ để quản lý nguyên liệu đầu vào và theo dõi các mẻ ủ rượu.
* Việc ghi nhận và theo dõi thu chi chủ yếu dựa trên kinh nghiệm, không có hệ thống cụ thể

### Những vấn đề còn tồn tại

Các vấn đề nếu trên ở hiện trạng sản xuất và kinh doanh khiến cho việc phát triển lâu dài gặp nhiều khó khan, không bắt kịp xu hướng hiện tại. Việc quản lý thủ công như vậy dẫn đến nhiều sai sót, thiếu chính xác trong việc kiểm soát tồn kho và lịch trình sản xuất, khó khan trong việc đánh giá hiệu quả kinh doanh, giảm chất lượng và sự đồng đều của sản phẩm. Tóm lại vấn đề hiện tại của các hộ sản xuất rượu truyền thống đang phải đối mặt bao gồm:

* Thiếu công cụ quản lý sổ sách, kho
* Khó khăn trong việc thống kê tài chính
* Khả năng cạnh tranh thấp do sản phẩm chưa đồng đều về chất lượng

### Những vấn đề cần tập trung nghiên cứu giải quyết

Đề tài “Xây dựng phần mềm quản lý sản xuất rượu RuKiSo” tập chung giải quyết các vấn đề được nêu trên bằng cách:

* **Hỗ trợ quản lý nguyên liệu và sản phẩm**: Cung cấp hệ thống nhập, xuất, thêm, sửa, xóa và theo dõi tồn kho của cả nguyên liệu và sản phẩm
* **Quản lý quy trình sản xuất**: Tạo công cụ theo dõi lịch trình sản xuất và nhắc nhở mốc thời gian quan trọng giúp đồng bộ chất lượng sản phẩm đầu ra
* **Thông kê và báo cáo tài chính**: Xây dựng hệ thống báo cáo doanh thu, chi phí, lợi nhuận, hỗ trợ người dùng đánh giá hiệu quả kinh doanh
* **Tăng cường tương tác người dùng**: Thiết kế giao diện thuần Việt, trức quan và dễ sử dụng

# Chương 2: Cơ sở lý thuyết và cơ sở thực nghiệm về vấn đề cần giải quyết

## 2.1. Cơ sở lý thuyết

### 2.1.1. Kiến trúc Client-Server

Mô hình client-server là một kiến trúc phổ biến trong phát triển phần mềm, nơi:

* **Client:** Đóng vai trò giao diện người dùng, chịu trách nhiệm gửi yêu cầu (request) và hiển thị dữ liệu nhận được từ server.
* **Server:** Đóng vai trò xử lý logic nghiệp vụ, quản lý cơ sở dữ liệu và cung cấp API để client tương tác.

### 2.1.2. .NET API

.NET API là một nền tảng mạnh mẽ để xây dựng các dịch vụ backend:

ASP.NET Core Web API: Được sử dụng để triển khai server-side. Đây là nền tảng mã nguồn mở, hiệu suất cao, hỗ trợ phát triển API RESTful.

Các điểm nổi bật:

* **Dependency Injection (DI):** Quản lý sự phụ thuộc giữa các thành phần, giúp code dễ kiểm thử và bảo trì.
* **Routing linh hoạt:** Cung cấp cơ chế định tuyến URL rõ ràng và dễ cấu hình.
* **Entity Framework Core:** Dùng để tương tác với cơ sở dữ liệu.

### 2.1.3. .NET MAUI

.NET MAUI (Multi-platform App UI) là nền tảng phát triển ứng dụng đa nền tảng (cross-platform) cho phép chia sẻ code giữa các hệ điều hành (Windows, Android, iOS, macOS):

* **MVVM (Model-View-ViewModel):** Là một mẫu thiết kế giúp tách biệt giao diện người dùng (View) khỏi logic nghiệp vụ (Model), thông qua một lớp trung gian gọi là ViewModel. Điều này giúp ứng dụng dễ bảo trì, mở rộng và kiểm thử.
* **Model**: Đại diện cho dữ liệu và logic nghiệp vụ, có thể bao gồm các lớp tương tác với cơ sở dữ liệu, API hoặc các dịch vụ khác. Model không chứa bất kì tham chiếu nào đến giao diện người dùng.
* **ViewModel**: Cầu nối giữa View và Model chưa dữ liệu mà view cần hiển thị, ở đây dữ liệu được sử lý logic tương tác, sự kiện hoặc gọi các dịch vụ và cập nhật trên view
* **View:** Thành phần giao diện người dùng kết nối với viewmodel thông qua cơ chế binding data, không chưa logic nghiệp vụ.
* **Data binding:** Là một cơ chế giúp dữ liệu từ view được cập nhât đến viewmodel hoặc ngược lại. Binding gồm nhiều mode và kiểu binding khác nhau giúp đơn giản hóa công việc.
* **Command binding:** Dùng để xử lý các sự kiện ở viewmodel thay vì viết trong code-behind của view điều này giúp tách biệt logic và giảm sự phụ thuộc của các thành phần trong một chương trình.

### 2.1.4. Cơ chế giao tiếp giữa client và server

Client và server giao tiếp qua giao thức HTTP/HTTPS bằng cách sử dụng các phương thức của RESTful API:

**GET**: Lấy dữ liệu (ví dụ: danh sách sản phẩm, nguyên liệu).

**POST**: Gửi dữ liệu mới (ví dụ: thêm giao dịch).

**PUT/PATCH**: Cập nhật dữ liệu (ví dụ: sửa thông tin sản phẩm).

**DELETE**: Xóa dữ liệu (ví dụ: xóa giao dịch).

### 2.1.5. Quản lý trạng thái với MVVM

Trong client sử dụng .NET MAUI, mô hình MVVM giúp:

* **Tách biệt logic và giao diện**: Dữ liệu được quản lý trong ViewModel thông qua các cơ chế binding.
* **Command và Event**: Thao tác người dùng (như bấm nút) được xử lý qua ICommand.
* **ObservableCollection**: Danh sách dữ liệu được cập nhật tự động mỗi khi có thay đổi, giúp giao diện luôn phản ánh trạng thái mới nhất.

## 2.2. Cơ sở thực nghiệm

### 2.2.1. Kiến trúc hệ thống

Hệ thống được chia thành hai phần chính:

* **Server:** Xây dựng bằng ASP.NET Core Web API được kết nối với cơ sở dữ liệu qua Entity Framework Core, dữ liệu được seed để kiểm tra các tính năng ngoài ra các API được thiết kế theo chuẩn RESTful, bao gồm các endpoint như:
  + /api/products: Quản lý sản phẩm.
  + /api/ingredients: Quản lý nguyên liệu.
  + /api/transactions: Quản lý giao dịch.
  + /api/batches: Quản lý nguyên liệu.
* **Client**: Xây dựng bằng .NET MAUI ap dụng mẫu thiết kế MVVM để quản lý giao diện và logic. Sử dụng HttpClient để gửi và nhận dữ liệu từ API. Các giao diện chính bao gồm:
  + Thông tin và danh sách sản phẩm (Products).
  + Thông tin và danh sách nguyên liệu (Ingredients).
  + Thông tin và danh sách các mẻ rượu đang ủ (Batches)
  + Quản lý giao dịch (Transactions).

### 2.2.2. Tương tác giữa client và server

* **Truyền dữ liệu**: Client gửi yêu cầu HTTP với JSON payload đến server, server phản hồi với dữ liệu JSON tương ứng.
* **Xử lý lỗi**: Client hiển thị popup thông báo lỗi khi có các vấn đề khi thêm sửa xóa dữ liệu

### 2.2.3. Môi trường thực nghiệm

Công nghệ sử dụng:

* **Server**: ASP.NET Core Web API, Entity Framework Core, SQL Server.
* **Client**: .NET MAUI.
* **Công cụ phát triển**: Visual Studio 2022, Git, SourceTree
* **Cơ sở dữ liệu**: SQL Server

Kiểm thử chức năng:

* Thêm, sửa, xóa sản phẩm từ client và kiểm tra dữ liệu trên server.
* Thêm, sửa, xóa nguyên liệu từ client và kiểm tra dữ liệu trên server.
* Thêm, sửa, xóa mẻ ủ rượu từ client và kiểm tra dữ liệu trên server.
* Thêm, sửa, xóa giao dịch từ client và kiểm tra dữ liệu trên server.
* Hiện thị các dữ liệu thống kê dưới dạng biểu đồ

### 2.2.4. Kết quả thực nghiệm

Client có giao diện đẹp mắt, dễ dàng sử dụng và thân thiện với người dùng. Ngoài ra còn hoạt động mượt mà, hiển thị đúng thông tin được gửi lên từ server.

Mô hình MVVM giúp dễ bảo trì và nâng cấp sau này

API phản hồi đúng chuẩn REST, cho phép client và server giao tiếp hiệu quả.

## 2.3 Lý do chọn công nghệ, framework

Là một trong nhưng tập đoàn công nghê lớn trong ngành công nghệ toàn cầu, Microsoft luôn luôn phát triển và hỗ trợ cho các công nghệ của mình một cách toàn diện nhất. Với hệ sinh thái rộng lớn và nền tảng công nghệ khổng lồ mà họ đang có, việc xây dựng phần mềm sử dụng các công nghệ do Microsoft phát triển và hỗ trợ trở nên dễ dàng hơn rất nhiều.

### 2.3.1 ASP.NET Core Web API cho server

ASP.NET Core Web API là một phần của ASP.NET Core, một framework mã nguồn mở, đa nền tảng được phát triển bởi Microsoft. ASP.NET Core Web API được sử dụng để xây dựng các dịch vụ HTTP RESTful, cho phép các ứng dụng khác nhau giao tiếp và trao đổi dữ liệu với nhau qua giao thức HTTP với các ưu điểm như:

* **Hiệu suất cao**: ASP.NET Core được tối ưu hóa để xử lý lượng lớn yêu cầu với thời gian phản hồi nhanh. Điều này quan trọng với các hệ thống client-server có khả năng mở rộng.
* **Đa nền tảng**: ASP.NET Core hoạt động trên Windows, Linux và macOS, cho phép triển khai server linh hoạt trong các môi trường khác nhau.
* **Hỗ trợ Dependency Injection (DI):** Đây là một tính năng tích hợp mạnh mẽ giúp giảm sự phụ thuộc giữa các thành phần, hỗ trợ tốt hơn cho các nguyên tắc SOLID và lập trình hướng đối tượng.
* **Tích hợp tốt với Entity Framework Core:** Cung cấp khả năng tương tác trực tiếp với cơ sở dữ liệu bằng cách sử dụng mô hình đối tượng (ORM), giúp giảm thiểu thời gian viết truy vấn SQL thủ công.
* **Hệ sinh thái phong phú:** ASP.NET Core hỗ trợ tích hợp các công cụ như Swagger (để tạo tài liệu API), Identity (quản lý xác thực người dùng), và SignalR (giao tiếp thời gian thực).
* **Bảo mật mạnh mẽ:** Tích hợp sẵn các cơ chế bảo mật như CORS, JWT, OAuth2, giúp bảo vệ API khỏi các lỗ hổng bảo mật phổ biến.

Ngoài các ưu điểm kể trên việc sử dụng ASP.NET Core Web API còn có một số lợi ích khác như có thể phát triển nhanh chóng nhờ các công cụ hỗ trợ và các thư việc mạnh mẽ đi kèm, có thể xây dựng các dịch vụ hỗ trợ tùy chỉnh với các yêu cầu khác nhau và cuối cùng nhưng quan trọng nhất là framework này là một sản phẩm của Microsoft nên có một cộng đồng hỗ trợ rất lớn và mạnh mẽ.

### 2.3.2 .NET MAUI và mẫu thiết kế MVVM cho client

**.NET MAUI (Multi-platform App UI)** là một nền tảng phát triển ứng dụng đa nền tảng hiện đại của Microsoft, được thiết kế để xây dựng các ứng dụng giao diện người dùng (UI) trên nhiều hệ điều hành khác nhau, bao gồm **Windows**, **Android**, **iOS**, và **macOS**. Là sự kế thừa của Xamarin.Forms, .NET MAUI không chỉ cải tiến mà còn tối ưu hóa khả năng phát triển ứng dụng, giúp lập trình viên dễ dàng chia sẻ code và tạo ra các trải nghiệm người dùng nhất quán trên mọi nền tảng. Được giới thiệu vào năm 2021 đây là một công nghệ rất mới nhưng đã được cộng đồng phát triển phần mềm đón nhận và yêu thích.

## 2.4 Vấn đề cần giải quyết

Trong quá trình nghiên cứu và phân tích hiện trạng, đề tài "Xây dựng phần mềm quản lý sản xuất rượu RuKiSo" hướng đến giải quyết những vấn đề cốt lõi sau đây:

### 2.4.1. Quản lý nguyên liệu và kho hàng

Việc quản lý nguyên liệu đầu vào như gạo nếp, men rượu, nước và các vật tư khác vẫn còn thực hiện theo cách thủ công, gây khó khăn trong việc theo dõi số lượng tồn kho và hạn sử dụng. Những vấn đề cần giải quyết:

* Tự động theo dõi tồn kho: Hệ thống cần cập nhật số lượng nguyên liệu mỗi khi có giao dịch nhập hoặc xuất.
* Quản lý hạn sử dụng nguyên liệu: Cảnh báo khi nguyên liệu sắp hết hạn để tránh lãng phí.

Dự báo nguyên liệu cần nhập: Hỗ trợ đưa ra quyết định nhập nguyên liệu dựa trên lịch sử tiêu thụ.

### 2.4.2. Quản lý quy trình sản xuất rượu

Việc sản xuất rượu truyền thống có nhiều giai đoạn như ủ men, lên men, chưng cất và bảo quản. Tuy nhiên, do không có công cụ hỗ trợ, nhiều hộ sản xuất không kiểm soát tốt các bước này, dẫn đến chất lượng sản phẩm không đồng đều. Những vấn đề cần giải quyết:

* Tạo và theo dõi các mẻ rượu: Mỗi mẻ rượu có ngày bắt đầu, dự kiến hoàn thành và trạng thái hiện tại.
* Nhắc nhở giai đoạn sản xuất: Hệ thống gửi thông báo khi đến thời điểm chuyển bước hoặc hoàn thành sản xuất.
* Ghi nhận thông tin sản xuất: Bao gồm nhiệt độ, thời gian lên men, sản lượng rượu thu được để tối ưu chất lượng.

### 2.4.3. Quản lý giao dịch bán hàng và tài chính

Việc ghi nhận doanh thu, chi phí và lợi nhuận chủ yếu dựa trên ghi chép sổ sách hoặc trí nhớ, dễ dẫn đến sai sót. Những vấn đề cần giải quyết:

* Ghi nhận doanh số bán hàng: Mỗi giao dịch được lưu trữ với thông tin chi tiết (ngày bán, khách hàng, số lượng, giá trị).
* Tính toán lợi nhuận: Tự động tính toán doanh thu, chi phí nguyên liệu và lợi nhuận theo từng tháng/quý.
* Xuất báo cáo tài chính: Cung cấp báo cáo doanh thu, lợi nhuận dưới dạng biểu đồ trực quan.

### 2.4.4. Cải thiện trải nghiệm người dùng với ứng dụng di động

Hệ thống hiện tại chủ yếu quản lý bằng giấy tờ hoặc file Excel trên máy tính. Điều này gây khó khăn cho người dùng khi cần kiểm tra thông tin nhanh chóng. Những vấn đề cần giải quyết:

* Giao diện thân thiện: Ứng dụng .NET MAUI giúp thiết kế UI dễ sử dụng cho người không quen công nghệ.
* Truy cập đa nền tảng: Hỗ trợ chạy trên cả Android, iOS và Windows để tiện lợi cho người dùng.
* Tích hợp tìm kiếm và bộ lọc: Hỗ trợ tìm kiếm sản phẩm, nguyên liệu, giao dịch nhanh chóng.

### 2.4.5. Tích hợp với hệ thống lưu trữ dữ liệu SQL Server

Với lượng dữ liệu ngày càng lớn, cần có một cơ sở dữ liệu mạnh mẽ và ổn định. Những vấn đề cần giải quyết:

* Tối ưu truy vấn dữ liệu: Sử dụng SQL Server để xử lý nhanh các truy vấn liên quan đến kho hàng, giao dịch.
* Đảm bảo an toàn dữ liệu: Cơ chế sao lưu định kỳ và phân quyền truy cập để bảo vệ dữ liệu quan trọng.
* Hỗ trợ mở rộng: Thiết kế hệ thống có thể dễ dàng mở rộng khi số lượng giao dịch tăng lên.

Tóm lại, các vấn đề cần giải quyết của đề tài bao gồm:

* Quản lý nguyên liệu và kho hàng – Theo dõi số lượng tồn kho, hạn sử dụng, dự báo nhập nguyên liệu.
* Quản lý quy trình sản xuất rượu – Ghi nhận từng giai đoạn sản xuất, nhắc nhở, tối ưu hóa quy trình.
* Quản lý giao dịch bán hàng và tài chính – Tự động ghi nhận doanh thu, chi phí, lợi nhuận và xuất báo cáo.
* Cải thiện trải nghiệm người dùng trên ứng dụng di động – Thiết kế UI trực quan, hỗ trợ tìm kiếm, truy cập đa nền tảng.
* Tích hợp với SQL Server để đảm bảo hiệu suất và bảo mật – Lưu trữ dữ liệu hiệu quả, hỗ trợ mở rộng hệ thống.