**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÙI TẤN PHÁT**

**ĐỀ CƯƠNG ĐỒ ÁN CHUYÊN NGÀNH**

**THÔNG TIN MÔI TRƯỜNG**

**TP. Hồ Chí Minh,24 tháng 2 năm 2025**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**TÊN ĐỀ TÀI**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ VÀ GIÁM SÁT**

**TÀI NGUYÊN THỦY SẢN TPHCM**

Giảng viên hướng dẫn : **ThS. Đặng Đức Trung**

Sinh viên thực hiện: **Bùi Tấn Phát**

Mã số sinh viên: **1050070020**

Lớp **: 10ĐH\_TTMT**

Khoá  **: 10**

**TP. Hồ Chí Minh, 24 tháng 2 năm 2025**

**MỞ ĐẦU**

**Cơ sở khoa học và tính thực tiễn của đề tài**

**Tính thực tiễn của đề tài**

Đề tài “Xây dựng phần mềm quản lý và giám sát tài nguyên thủy sản TP.HCM” mang ý nghĩa thực tiễn lớn trong bối cảnh ngành thủy sản tại TP.HCM đang đối mặt với nhiều thách thức về quản lý và bảo vệ tài nguyên. TP.HCM, với hệ thống sông ngòi và vùng ven biển phong phú, là một trong những trung tâm khai thác và nuôi trồng thủy sản lớn của Việt Nam. Theo báo cáo năm 2023 của Sở Nông nghiệp và Phát triển nông thôn TP.HCM, sản lượng thủy sản đạt hơn 60.000 tấn mỗi năm, nhưng việc quản lý tài nguyên thủy sản vẫn chủ yếu dựa vào các phương pháp thủ công, dẫn đến khó khăn trong việc theo dõi, giám sát và ngăn chặn khai thác quá mức.

Phần mềm quản lý và giám sát tài nguyên thủy sản được phát triển nhằm tự động hóa quy trình quản lý, cung cấp một nền tảng tích hợp để theo dõi số lượng, chất lượng và tình trạng tài nguyên thủy sản tại TP.HCM. Hệ thống này giúp các cơ quan chức năng, như Sở Nông nghiệp và Phát triển nông thôn, Chi cục Thủy sản, và các hợp tác xã, dễ dàng quản lý dữ liệu về khu vực khai thác, sản lượng, loài thủy sản, và các hoạt động vi phạm. Đặc biệt, phần mềm hỗ trợ giám sát thời gian thực, giúp phát hiện sớm các hành vi khai thác bất hợp pháp và bảo vệ đa dạng sinh học.

Phần mềm cũng mang lại lợi ích kinh tế - xã hội rõ rệt. Bằng cách cung cấp dữ liệu chính xác và kịp thời, hệ thống hỗ trợ các cơ quan ra quyết định chiến lược, từ đó tối ưu hóa hoạt động khai thác, nuôi trồng và bảo tồn. Ví dụ, việc giám sát số lượng cá giống thả nuôi có thể giúp điều chỉnh kế hoạch sản xuất, tránh tình trạng thiếu hụt hoặc dư thừa. Ngoài ra, phần mềm còn cải thiện trải nghiệm của các bên liên quan, như ngư dân và doanh nghiệp thủy sản, thông qua quy trình đăng ký và báo cáo trực tuyến, giảm thiểu thủ tục hành chính.

Về mặt bảo mật, hệ thống sử dụng MongoDB với các cơ chế mã hóa và kiểm soát truy cập, đảm bảo dữ liệu nhạy cảm (như thông tin ngư dân hoặc báo cáo vi phạm) được bảo vệ, tuân thủ Luật An ninh mạng Việt Nam (2018). Phân tích dữ liệu tích hợp trong phần mềm cũng cho phép tạo báo cáo chi tiết về xu hướng khai thác, từ đó hỗ trợ xây dựng các chính sách dài hạn cho ngành thủy sản.

Tóm lại, phần mềm không chỉ đáp ứng nhu cầu chuyển đổi số trong quản lý tài nguyên thủy sản mà còn góp phần thúc đẩy phát triển bền vững, bảo vệ môi trường và nâng cao hiệu quả kinh tế tại TP.HCM.

**Cơ sở khoa học của đề tài**

Sự phát triển của công nghệ thông tin, đặc biệt là các nền tảng web và cơ sở dữ liệu NoSQL, đã tạo điều kiện cho việc xây dựng các hệ thống quản lý tài nguyên phức tạp. Đề tài này dựa trên sự giao thoa giữa công nghệ thông tin và quản lý tài nguyên thiên nhiên, tận dụng các công nghệ hiện đại như Node.js, React.js và MongoDB để phát triển một hệ thống mạnh mẽ, linh hoạt và dễ mở rộng.

Phần mềm sử dụng Node.js cho backend, một môi trường runtime hiệu suất cao, phù hợp để xử lý các yêu cầu đồng thời từ nhiều người dùng, như ngư dân đăng ký hoặc cán bộ giám sát dữ liệu. React.js, với khả năng xây dựng giao diện người dùng động và đáp ứng, đảm bảo trải nghiệm mượt mà trên cả web và thiết bị di động. MongoDB, một cơ sở dữ liệu NoSQL, được chọn vì khả năng lưu trữ dữ liệu phi cấu trúc, lý tưởng cho các tập dữ liệu đa dạng như thông tin loài thủy sản, khu vực khai thác, và báo cáo giám sát.

Dự án áp dụng phương pháp Agile, cho phép phát triển lặp lại và phản hồi liên tục, đảm bảo phần mềm đáp ứng đúng nhu cầu thực tế. Các chức năng chính của hệ thống bao gồm:

* Quản lý tài nguyên thủy sản: Theo dõi thông tin về loài, khu vực khai thác, và sản lượng.
* Giám sát thời gian thực: Phát hiện và ghi nhận các hoạt động khai thác bất hợp pháp.
* Báo cáo và phân tích: Tạo báo cáo chi tiết về tình trạng tài nguyên thủy sản.

**Công nghệ áp dụng:**

* Backend: Node.js
* Frontend: React.js
* Cơ sở dữ liệu: MongoDB
* Công cụ phát triển: Visual Studio Code, Android Studio (cho phát triển ứng dụng di động trong tương lai)

**Tính ứng dụng:**

* Tăng hiệu quả quản lý tài nguyên, giảm thời gian xử lý thủ công.
* Cải thiện khả năng giám sát, hỗ trợ bảo vệ đa dạng sinh học.
* Dễ dàng mở rộng để tích hợp thêm các tính năng như AI phân tích dữ liệu hoặc ứng dụng di động.

**Mục đích nghiên cứu**

Mục đích nghiên cứu đề tài là tạo ra phần mềm theo dõi và quản lí việc sử dụng điên năng lượng mặt trời tại TP.HCM:

* Hiển thị dữ liệu thời gian thực về sản lượng điện từ các tấm pin mặt trời, tiêu thụ điện năng trong gia đình/công ty.
* Cho phép người dùng cấu hình, tối ưu hóa việc sử dụng điện (ví dụ: điều chỉnh điện lưu trữ, báo cáo tiết kiệm năng lượng).
* Xem xét lịch sử dữ liệu, so sánh hiệu suất theo tháng/năm, dự báo nhu cầu điện.

**Đối tượng và phạm vi nghiên cứu:**

Đối tượng: Các khu vực nuôi trồng và khai thác thủy sản tại TP. Hồ Chí Minh, cơ quan quản lý tài nguyên thủy sản, người dân tham gia sản xuất thủy sản.

Phạm vi: Tập trung vào việc giám sát sản lượng, chất lượng nước và tình trạng môi trường tại các khu vực trọng điểm

* **Các ngôn ngữ lập trình:**

Ngôn ngữ Dart: Dart là ngôn ngữ chính của Flutter, dùng để xây dựng ứng dụng đa nền tảng (mobile, web, desktop).

* **Các công cụ:**

Visual Studio Code (VS Code): Trình soạn thảo mã nguồn hỗ trợ lập trình Dart và Flutter.

Android Studio: Môi trường phát triển tích hợp để lập trình và kiểm thử ứng dụng Android.

Flutter: Framework xây dựng ứng dụng đa nền tảng với một codebase duy nhất.

**Phương pháp nghiên cứu**

* **Phương pháp nghiên cứu lý thuyết:**

Nghiên cứu tài liệu liên quan đến quản lý tài nguyên thủy sản: Tìm hiểu về nuôi trồng, khai thác thủy sản, các yếu tố môi trường ảnh hưởng (chất lượng nước, ô nhiễm).

Nghiên cứu phân tích và thiết kế hệ thống: Xác định mục tiêu, chức năng, giao diện và kiến trúc phần mềm.

Nghiên cứu thị trường: Đánh giá nhu cầu sử dụng phần mềm quản lý thủy sản tại TP. Hồ Chí Minh.

Nghiên cứu hiệu năng: Tối ưu phần mềm để chạy hiệu quả trên các thiết bị di động.

Phương pháp thực nghiệm:

Thu thập và phân tích dữ liệu thực tế: Lấy dữ liệu từ các khu vực nuôi trồng/khai thác thủy sản tại TP. Hồ Chí Minh (sản lượng, chất lượng nước).

Mô phỏng và thử nghiệm phần mềm: Kiểm tra phần mềm với dữ liệu thử nghiệm để đánh giá hiệu năng.

Khảo sát và thu thập phản hồi từ người dùng (cơ quan quản lý, người dân).

Đánh giá hiệu quả quản lý tài nguyên thủy sản.

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành đến thầy ThS. Đặng Đức Trung, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án. Những góp ý quý báu của thầy đã giúp em hoàn thiện đề tài một cách hệ thống và khoa học.

Em cũng xin cảm ơn Trường Đại học Tài nguyên và Môi trường TP.HCM, Khoa Công nghệ Thông tin, cùng các thầy cô đã cung cấp môi trường học tập và tài liệu hỗ trợ, tạo điều kiện thuận lợi để em hoàn thành đồ án.

Do thời gian và kinh nghiệm còn hạn chế, báo cáo không tránh khỏi thiếu sót. Em rất mong nhận được ý kiến đóng góp từ thầy để tiếp tục hoàn thiện kiến thức và kỹ năng.

Em xin chân thành cảm ơn!

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

|  |  |
| --- | --- |
|  | …………..,ngày….tháng….năm……  NGƯỜI NHẬN XÉT  *(ký tên)* |

**NHẬN XÉT**

**(Của giảng viên phản biện 1)**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

|  |
| --- |
| ………….., ngày….tháng….năm……  NGƯỜI NHẬN XÉT  *(ký tên)* |

**NHẬN XÉT**

**(Của giảng viên phản biện 2 )**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………….., ngày….tháng….năm……

NGƯỜI NHẬN XÉT

(ký tên)

Mục Lục

[Chương 1. TỔNG QUAN TÀI LIỆU 13](#_Toc197287574)

[1.1. Tổng quan về Node.js 13](#_Toc197287575)

[Khái niệm 13](#_Toc197287576)

[Lợi ích 13](#_Toc197287577)

[1.2. Tổng quan về React.js 14](#_Toc197287578)

[Khái niệm 14](#_Toc197287579)

[Lợi ích 14](#_Toc197287580)

[1.3. Tổng quan về MongoDB 15](#_Toc197287581)

[Khái niệm 15](#_Toc197287582)

[Lợi ích 16](#_Toc197287583)

[1.4. Tổng quan về phần mềm quản lý tài nguyên 16](#_Toc197287584)

[Các phần mềm hiện có 16](#_Toc197287585)

[Đánh giá ưu và nhược điểm 16](#_Toc197287586)

[Chương 2: PHƯƠNG PHÁP THỰC HIỆN 17](#_Toc197287587)

[2.1. Phương pháp nghiên cứu 17](#_Toc197287588)

[2.2. Đối tượng nghiên cứu 17](#_Toc197287589)

[2.3. Công cụ và kỹ thuật 17](#_Toc197287590)

[2.4. Quy trình thực hiện 18](#_Toc197287591)

[2.5. Quản lý rủi ro 18](#_Toc197287592)

[Chương 3. CÀI ĐẶT THỰC NGHIỆM 19](#_Toc197287593)

[3.1. Phân tích và thiết kế hệ thống 19](#_Toc197287594)

[3.2Sơ đồ Use Case tổng quát 22](#_Toc197287595)

[3.3Đặc tả Use Case 23](#_Toc197287596)

[3.4Sơ đồ DFD 24](#_Toc197287597)

[3.5Giao diện app 26](#_Toc197287598)

[3.6Giao diện phần quyền admin quản trị 27](#_Toc197287599)

[3.7Giao diện chỉnh sửa thông tin của admin 28](#_Toc197287600)

[3.8Giao diện quản lý tài nguyên 30](#_Toc197287601)

[3.9Giao diện phê duyệt của admin 37](#_Toc197287602)

[3.10Giao diện cá nhân 38](#_Toc197287603)

[3.11Giao diện của ngư dân quản lý 39](#_Toc197287604)

[3.12Giao diện đăng ký quyền thông qua admin 40](#_Toc197287605)

[3.13 Thông tin người dùng 41](#_Toc197287606)

[CHƯƠNG 4 KẾT LUẬN 42](#_Toc197287607)

* 1. **Thời gian thực hiện**

Từ 24/2/2025 đến 30/5/2025

* 1. **Kết quả dự kiến đạt được**
* Thiết kế được phần mềm thiết kế hệ thống quản lý thủy sản TPHCM
* Viết được khóa luận theo yêu cầu của Khoa Công nghệ thông tin, Trường Đại Học Tài nguyên Và Môi trường TP. Hồ Chí Minh.

1. **Nội dung**

# Chương 1. TỔNG QUAN TÀI LIỆU

## 1.1. Tổng quan về Node.js

### Khái niệm

Node.js là một môi trường runtime mã nguồn mở, cho phép chạy JavaScript bên ngoài trình duyệt, được phát triển bởi Ryan Dahl vào năm 2009. Dựa trên engine V8 của Google Chrome, Node.js nổi bật với khả năng xử lý bất đồng bộ, phù hợp cho các ứng dụng yêu cầu hiệu suất cao như hệ thống quản lý tài nguyên thủy sản.

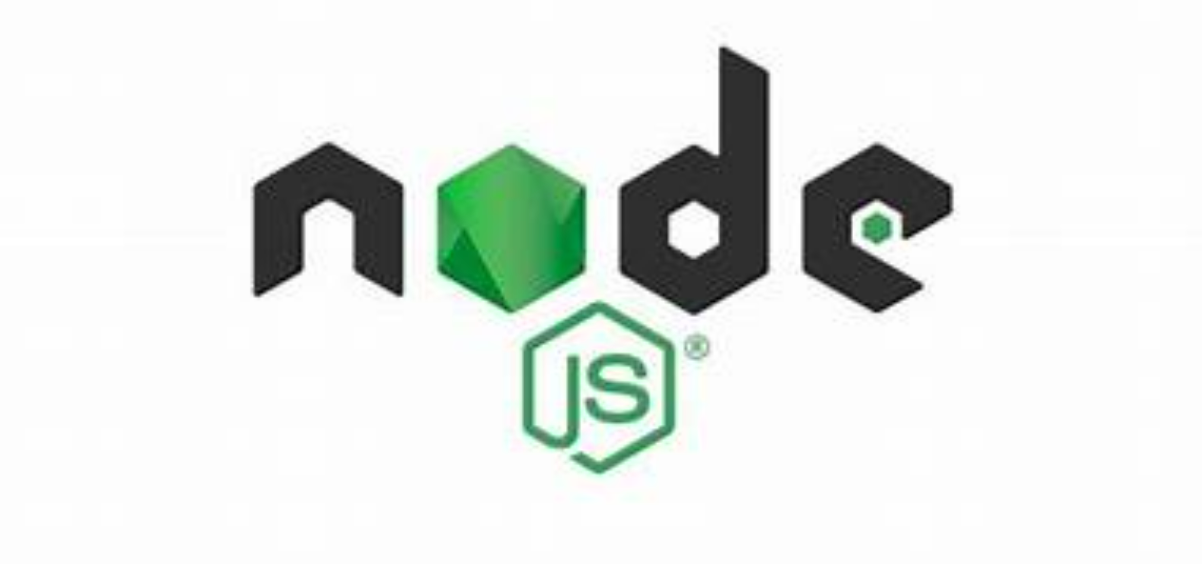
Node.js sử dụng mô hình I/O không chặn, giúp xử lý hàng nghìn yêu cầu đồng thời mà không gây tắc nghẽn. Trong dự án này, Node.js được sử dụng để xây dựng backend, xử lý các tác vụ như quản lý dữ liệu thủy sản, giám sát thời gian thực, và tạo báo cáo.

### Lợi ích

* **Hiệu suất cao**: Xử lý nhanh các yêu cầu đồng thời.
* **Cộng đồng lớn**: Hỗ trợ nhiều thư viện qua npm, như Express.js cho API.
* **Dễ mở rộng**: Phù hợp cho các hệ thống phân tán.

**Ứng dụng trong quản lý tài nguyên**

Node.js được sử dụng rộng rãi trong các hệ thống giám sát, như hệ thống quản lý nông nghiệp hoặc môi trường. Ví dụ, nền tảng IoT nông nghiệp của IBM sử dụng Node.js để thu thập và xử lý dữ liệu thời gian thực, tương tự cách hệ thống này giám sát tài nguyên thủy sản.



## 1.2. Tổng quan về React.js

### Khái niệm

React.js là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook vào năm 2013, dùng để xây dựng giao diện người dùng động và đáp ứng. React.js sử dụng mô hình component-based, giúp phát triển giao diện nhanh chóng và tái sử dụng mã.

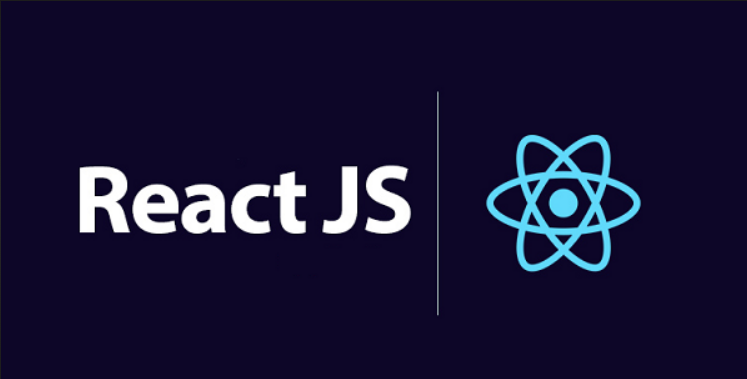
Trong dự án, React.js được dùng để xây dựng frontend, cung cấp giao diện cho cán bộ quản lý và ngư dân để nhập liệu, xem báo cáo, và theo dõi tài nguyên thủy sản.

### Lợi ích

* **Giao diện đáp ứng**: Tối ưu cho cả web và mobile.
* **Tái sử dụng component**: Giảm thời gian phát triển.
* **Hiệu suất cao**: Sử dụng Virtual DOM để tối ưu hóa render.

**Ứng dụng**

React.js được sử dụng trong nhiều nền tảng quản lý, như hệ thống quản lý nông nghiệp của John Deere, nơi giao diện người dùng cần hiển thị dữ liệu phức tạp và tương tác cao.



## 1.3. Tổng quan về MongoDB

### Khái niệm

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở, lưu trữ dữ liệu dưới dạng tài liệu JSON, phù hợp cho các ứng dụng yêu cầu linh hoạt trong cấu trúc dữ liệu. MongoDB được chọn cho dự án này vì khả năng xử lý dữ liệu phi cấu trúc, như thông tin loài thủy sản hoặc báo cáo giám sát.

### Lợi ích

* **Linh hoạt**: Hỗ trợ dữ liệu không đồng nhất.
* **Khả năng mở rộng**: Dễ dàng mở rộng theo chiều ngang.
* **Bảo mật**: Hỗ trợ mã hóa và kiểm soát truy cập.

**So sánh với các DBMS khác**

So với SQL Server, MongoDB phù hợp hơn cho dữ liệu phi cấu trúc và yêu cầu mở rộng nhanh. MySQL tuy tiết kiệm chi phí nhưng kém linh hoạt hơn trong xử lý dữ liệu phức tạp.

## 1.4. Tổng quan về phần mềm quản lý tài nguyên

### Các phần mềm hiện có

* **Quốc tế**:
  + **Triton (OceanMind)**: Hệ thống giám sát thủy sản toàn cầu, sử dụng AI và dữ liệu vệ tinh.
  + **Global Fishing Watch**: Theo dõi hoạt động khai thác thủy sản thời gian thực.
* **Trong nước**:
  + **Hệ thống VNFishbase**: Quản lý cơ sở dữ liệu thủy sản, nhưng thiếu tính năng giám sát thời gian thực.
  + **Phần mềm của Chi cục Thủy sản**: Chủ yếu quản lý hành chính, chưa tích hợp công nghệ hiện đại.

### Đánh giá ưu và nhược điểm

* **Ưu điểm**: Các hệ thống quốc tế cung cấp giám sát thời gian thực và phân tích dữ liệu mạnh mẽ. Các hệ thống trong nước dễ tiếp cận và chi phí thấp.
* **Nhược điểm**: Hệ thống quốc tế có chi phí cao, không tối ưu cho bối cảnh Việt Nam. Hệ thống trong nước thiếu tích hợp công nghệ hiện đại và khả năng mở rộng.



# Chương 2: PHƯƠNG PHÁP THỰC HIỆN

## 2.1. Phương pháp nghiên cứu

* **Phân tích định tính**: Phỏng vấn cán bộ Chi cục Thủy sản và ngư dân để xác định yêu cầu, như quản lý dữ liệu loài, giám sát khai thác, và báo cáo.
* **Phân tích định lượng**: Đo lường hiệu suất hệ thống, như thời gian phản hồi API và tốc độ truy vấn MongoDB.
* **Nghiên cứu tài liệu**: Tham khảo tài liệu về Node.js, React.js, MongoDB và các hệ thống quản lý tài nguyên thủy sản.

## 2.2. Đối tượng nghiên cứu

* **Phần mềm quản lý tài nguyên thủy sản**: Bao gồm các tính năng như quản lý dữ liệu, giám sát, và báo cáo.
* **Người dùng chính**:
  + **Cán bộ quản lý**: Quản lý dữ liệu và giám sát hoạt động khai thác.
  + **Ngư dân**: Đăng ký và báo cáo thông tin khai thác.
* **Công nghệ**: Node.js, React.js, MongoDB.

## 2.3. Công cụ và kỹ thuật

* **Framework**: Express.js (Node.js) cho backend, React.js cho frontend.
* **Công cụ phát triển**: Visual Studio Code, Android Studio.
* **Cơ sở dữ liệu**: MongoDB.
* **Giao diện**: Tailwind CSS cho thiết kế đáp ứng.

## 2.4. Quy trình thực hiện

**Phân tích yêu cầu**: Xác định chức năng chính (quản lý, giám sát, báo cáo).

**Thiết kế hệ thống**: Xây dựng mô hình dữ liệu MongoDB và kiến trúc API.

**Phát triển phần mềm**: Viết mã backend (Node.js), frontend (React.js), tích hợp MongoDB.

**Kiểm thử**: Kiểm thử đơn vị, tích hợp, và người dùng.

**Triển khai**: Triển khai trên Heroku hoặc máy chủ cục bộ.

## 2.5. Quản lý rủi ro

* **Tắc nghẽn cơ sở dữ liệu**: Tối ưu hóa truy vấn MongoDB.
* **Lỗi API**: Triển khai cơ chế retry và fallback.
* **Khó khăn sử dụng**: Cung cấp tài liệu hướng dẫn và đào tạo.

**2.6. Phạm vi và giới hạn**

* **Phạm vi**: Quản lý và giám sát tài nguyên thủy sản tại TP.HCM, triển khai trên web.
* **Giới hạn**: Chưa tích hợp AI phân tích hoặc ứng dụng di động.
* **Hướng phát triển**: Thêm tính năng AI, phát triển ứng dụng di động.

# Chương 3. CÀI ĐẶT THỰC NGHIỆM

## 3.1. Phân tích và thiết kế hệ thống

**Sơ đồ Usecase**

**Danh sách Actor**

|  |  |  |
| --- | --- | --- |
| STT | Actor | Diễn giải |
| 1 | Quản lý | Quản lý dữ liệu, phân quyền, giám sát, tạo báo cáo. |
| 2 | Ngư dân | Đăng ký thông tin, báo cáo khai thác, xem dữ liệu công khai. |

Quản lý phân quyền

* Thêm quyền mới: Chức năng này chỉ dành cho Admin, nhằm đảm bảo việc phân quyền được kiểm soát chặt chẽ.

Quản lý sự kiện: Các chức năng liên quan đến sự kiện đều có thể được thực hiện bởi cả Admin và Nhân viên:

* Thêm sự kiện
* Cập nhật sự kiện
* Xóa sự kiện
* Xem sự kiện
* Xem danh sách sự kiện

Nhân viên có thể sử dụng các chức năng này để phục vụ công việc quản lý và đăng ký sự kiện theo yêu cầu khách hàng.

Quản lý tài khoản: Các chức năng này chỉ dành riêng cho Admin nhằm đảm bảo tính bảo mật và toàn vẹn của hệ thống người dùng:

* Thêm tài khoản
* Cập nhật thông tin tài khoản
* Xóa tài khoản
* Xem danh sách tài khoản

**Sơ đồ Usecase**

**Danh sách Actor**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Use-case | Actor | Mô tả |
|  | Quản lý phân quyền | Quản lý | Thêm, sửa, xóa quyền cho các tài khoản trong hệ thống. |
|  | Quản lý dữ liệu thủy sản | Quản lý | Thêm, sửa, xóa, xem thông tin loài, khu vực, sản lượng thủy sản. |
|  | Giám sát khai thác | Quản lý | Theo dõi và cảnh báo các hoạt động khai thác bất hợp pháp. |
|  | Tạo báo cáo | Quản lý | Tạo báo cáo về sản lượng, khu vực, và xu hướng khai thác. |
|  | Đăng ký thông tin | Ngư dân | Đăng ký thông tin cá nhân và tàu thuyền. |
|  | Báo cáo khai thác | Ngư dân | Báo cáo sản lượng khai thác hàng ngày/tháng. |
|  | Xem dữ liệu công khai | Ngư dân | Xem thông tin công khai về tài nguyên thủy sản. |
|  | Quản lý tài khoản | Quản lý | Thêm, sửa, xóa, xem danh sách tài khoản người dùng. |

## 3.2Sơ đồ Use Case tổng quát

**Mô tả**:

* **Tác nhân (Actor)**:
  + **Quản lý**: Người quản trị hệ thống, có quyền truy cập đầy đủ các chức năng.
  + **Ngư dân**: Người dùng cơ bản, chỉ thực hiện các chức năng liên quan đến đăng ký và báo cáo.
* **Use Case chính**:
  + Quản lý phân quyền, quản lý dữ liệu thủy sản, giám sát khai thác, tạo báo cáo (Quản lý).
  + Đăng ký thông tin, báo cáo khai thác, xem dữ liệu công khai (Ngư dân).
  + Đăng nhập (chung cho cả Quản lý và Ngư dân).

**Use Case phân rã**

**a. Quản lý tài khoản**

**Mô tả**:

* **Actor**: Quản lý
* **Use Case con**:
  + Đăng nhập
  + Thêm tài khoản
  + Sửa tài khoản
  + Xóa tài khoản
  + Xem danh sách tài khoản
* **Mối quan hệ**:
  + “Quản lý tài khoản” bao gồm (include) các use case con.
  + Đăng nhập là điều kiện tiên quyết cho tất cả các hành động.

## 3.3Đặc tả Use Case

**Use Case: Đăng ký thông tin**

* **Actor**: Ngư dân
* **Mô tả**: Ngư dân đăng ký thông tin cá nhân và tàu thuyền để tham gia hệ thống.
* **Điều kiện tiên quyết**: Người dùng chưa có tài khoản.
* **Luồng chính**:
  1. Ngư dân truy cập trang đăng ký.
  2. Nhập thông tin (tên, số điện thoại, số đăng ký tàu, v.v.).
  3. Hệ thống kiểm tra tính hợp lệ (định dạng, trùng lặp).
  4. Nếu hợp lệ, lưu thông tin vào MongoDB.
  5. Hiển thị thông báo đăng ký thành công.

## 3.4Sơ đồ DFD

**Sơ đồ DFD mức 0**

**Mô tả**:

* **Tác nhân ngoài**: Quản lý, Ngư dân
* **Tiến trình chính**: Hệ thống quản lý và giám sát tài nguyên thủy sản
* **Luồng dữ liệu**:
  + Quản lý gửi: Yêu cầu quản lý dữ liệu, giám sát, tạo báo cáo, quản lý tài khoản.
  + Ngư dân gửi: Yêu cầu đăng ký, báo cáo khai thác, xem dữ liệu công khai.
  + Hệ thống trả về: Kết quả quản lý, báo cáo, thông báo đăng ký, dữ liệu công khai.
* **Kho dữ liệu**: FisheryResources, Users

**Sơ đồ DFD mức 1**

**Mô tả**:

* **Tiến trình con**:
  + 1.1: Quản lý tài khoản
  + 1.2: Quản lý dữ liệu thủy sản
  + 1.3: Giám sát khai thác
  + 1.4: Tạo báo cáo
  + 1.5: Đăng ký và báo cáo (Ngư dân)
* **Luồng dữ liệu**:
  + Quản lý gửi yêu cầu đến các tiến trình 1.1, 1.2, 1.3, 1.4.
  + Ngư dân gửi yêu cầu đến tiến trình 1.5.
  + Dữ liệu được lưu/truy xuất từ kho “FisheryResources” và “Users”.
* **Kho dữ liệu**: FisheryResources, Users

**Lược đồ Activity**

**Activity: Đăng ký thông tin**

**Mô tả**:

* **Bắt đầu**: Ngư dân truy cập trang đăng ký.
* **Hiển thị form**: Hệ thống hiển thị form đăng ký.
* **Nhập thông tin**: Ngư dân nhập tên, số điện thoại, số tàu, v.v.
* **Kiểm tra hợp lệ**: Hệ thống kiểm tra định dạng và trùng lặp.
* **Rẽ nhánh**:
  + Hợp lệ: Lưu thông tin, thông báo thành công, kết thúc.
  + Không hợp lệ: Yêu cầu nhập lại, quay lại bước nhập thông tin.

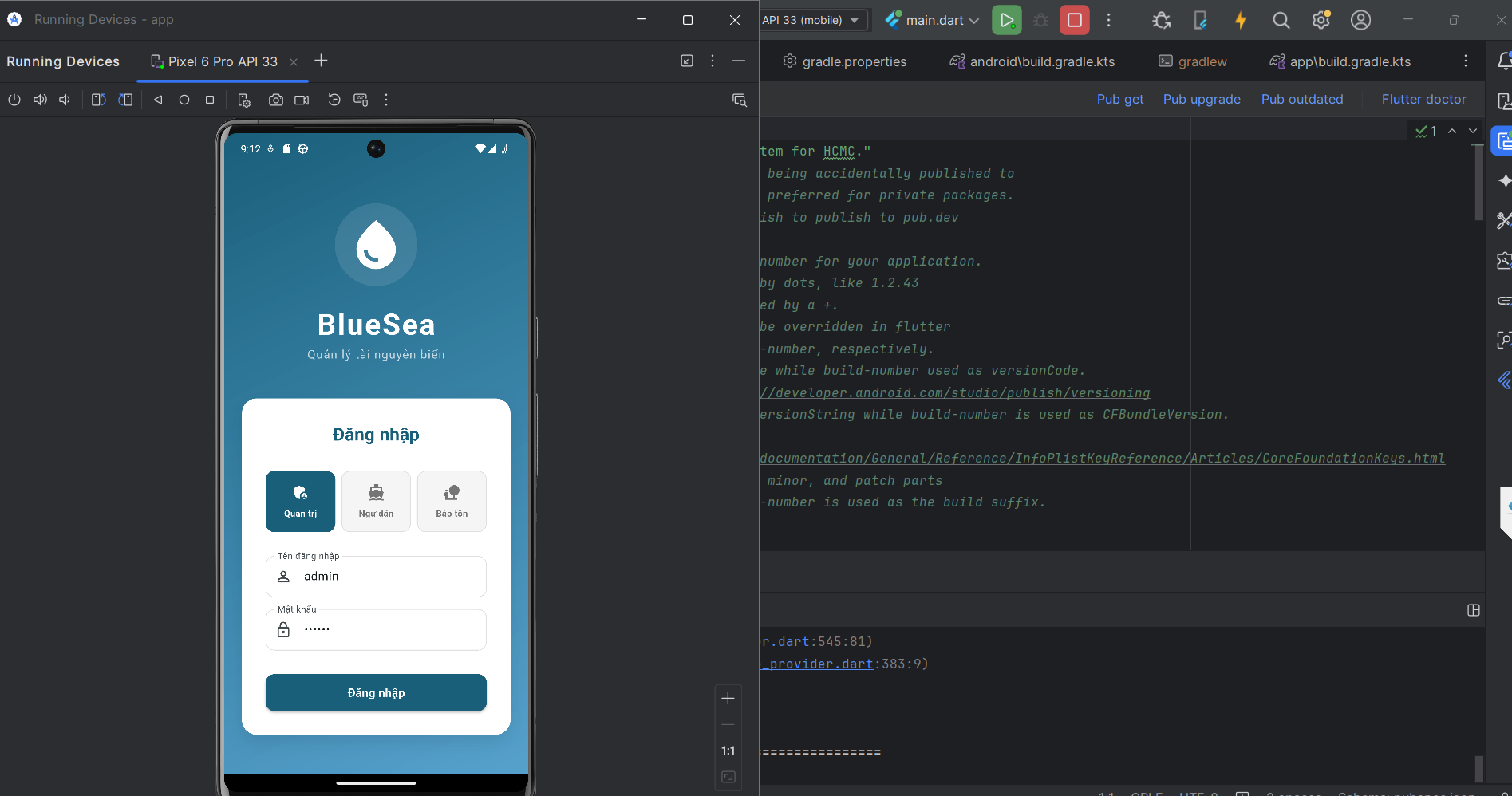
**Activity: Giám sát khai thác**

**Mô tả**:

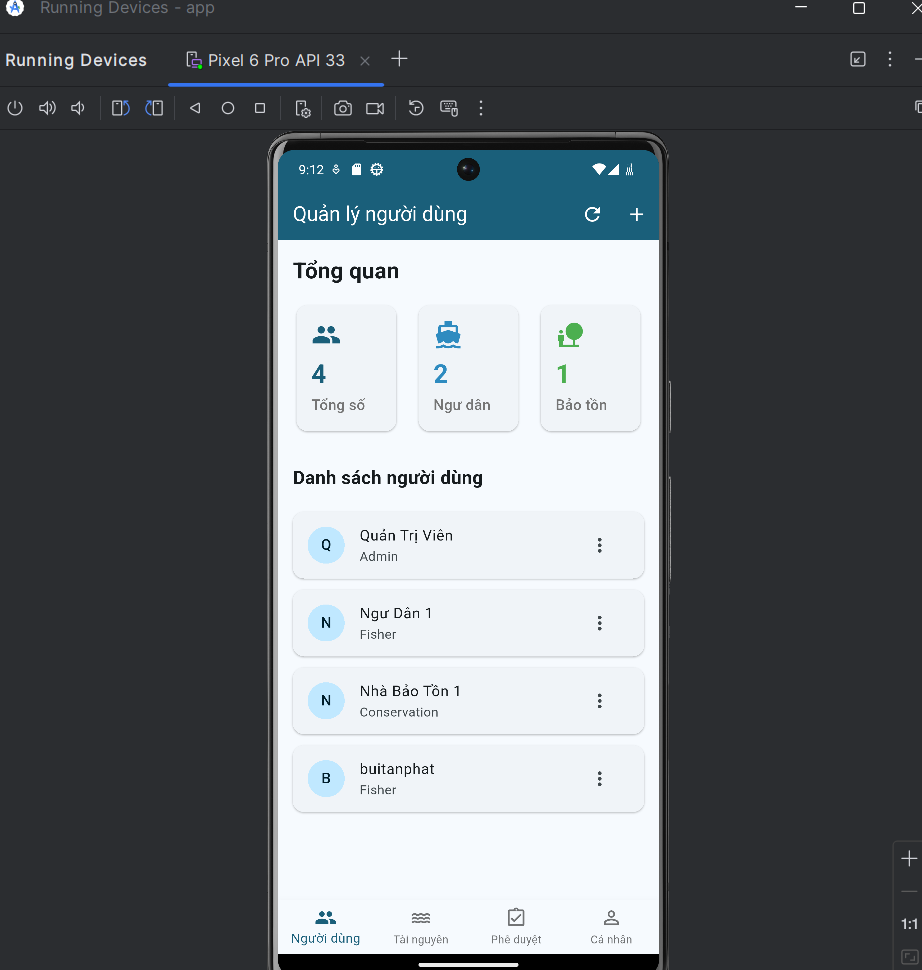
* **Bắt đầu**: Quản lý truy cập giao diện giám sát.
* **Hiển thị dữ liệu**: Hệ thống hiển thị sản lượng, khu vực.
* **Kiểm tra bất thường**: Hệ thống so sánh với ngưỡng cho phép.
* **Rẽ nhánh**:
  + Có bất thường: Gửi cảnh báo, ghi nhận vi phạm, kết thúc.
  + Không bất thường: Tiếp tục giám sát

## 3.5Giao diện app

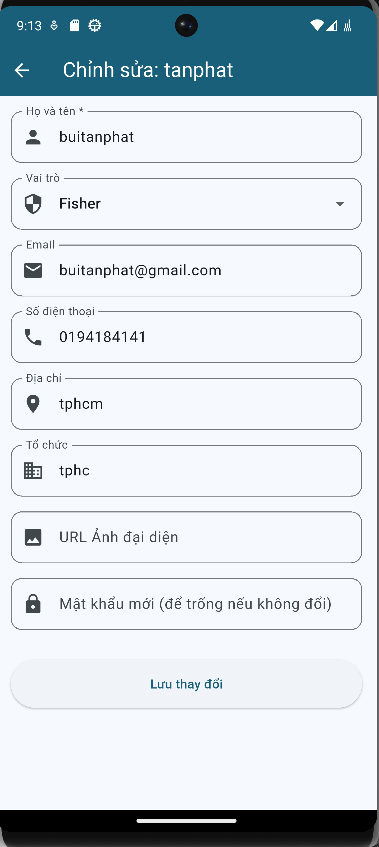
Màn hình main



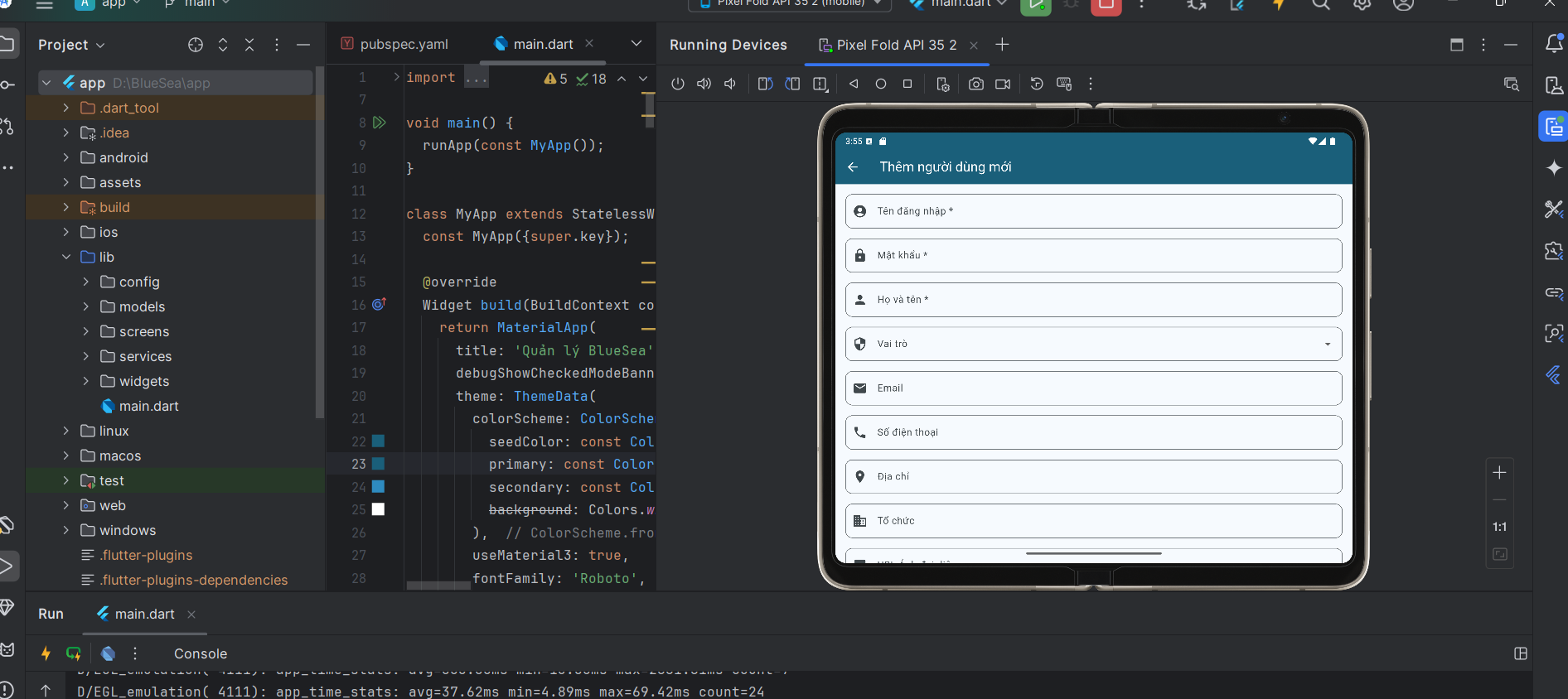
## 3.6Giao diện phần quyền admin quản trị



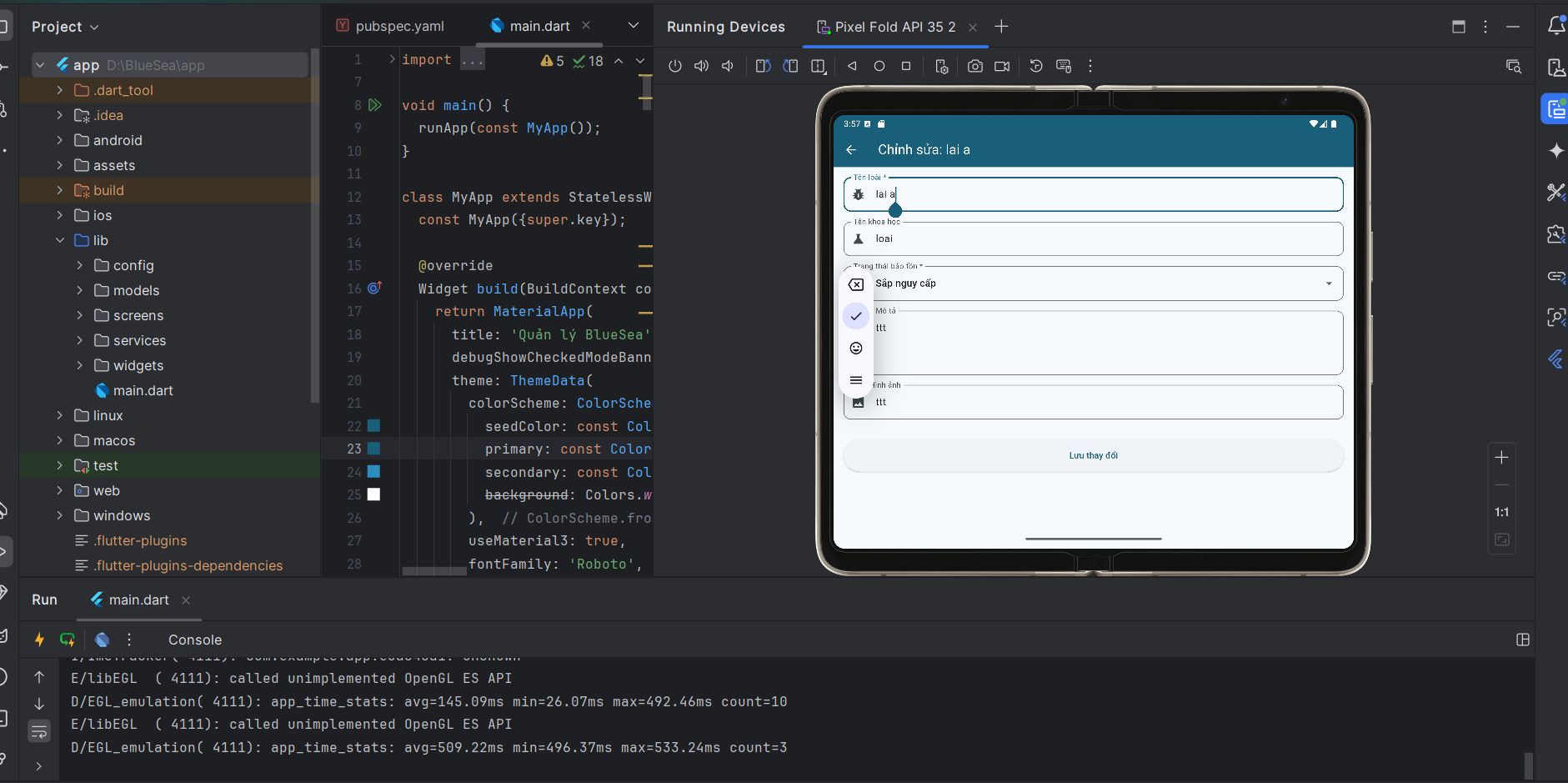
## 3.7Giao diện chỉnh sửa thông tin của admin



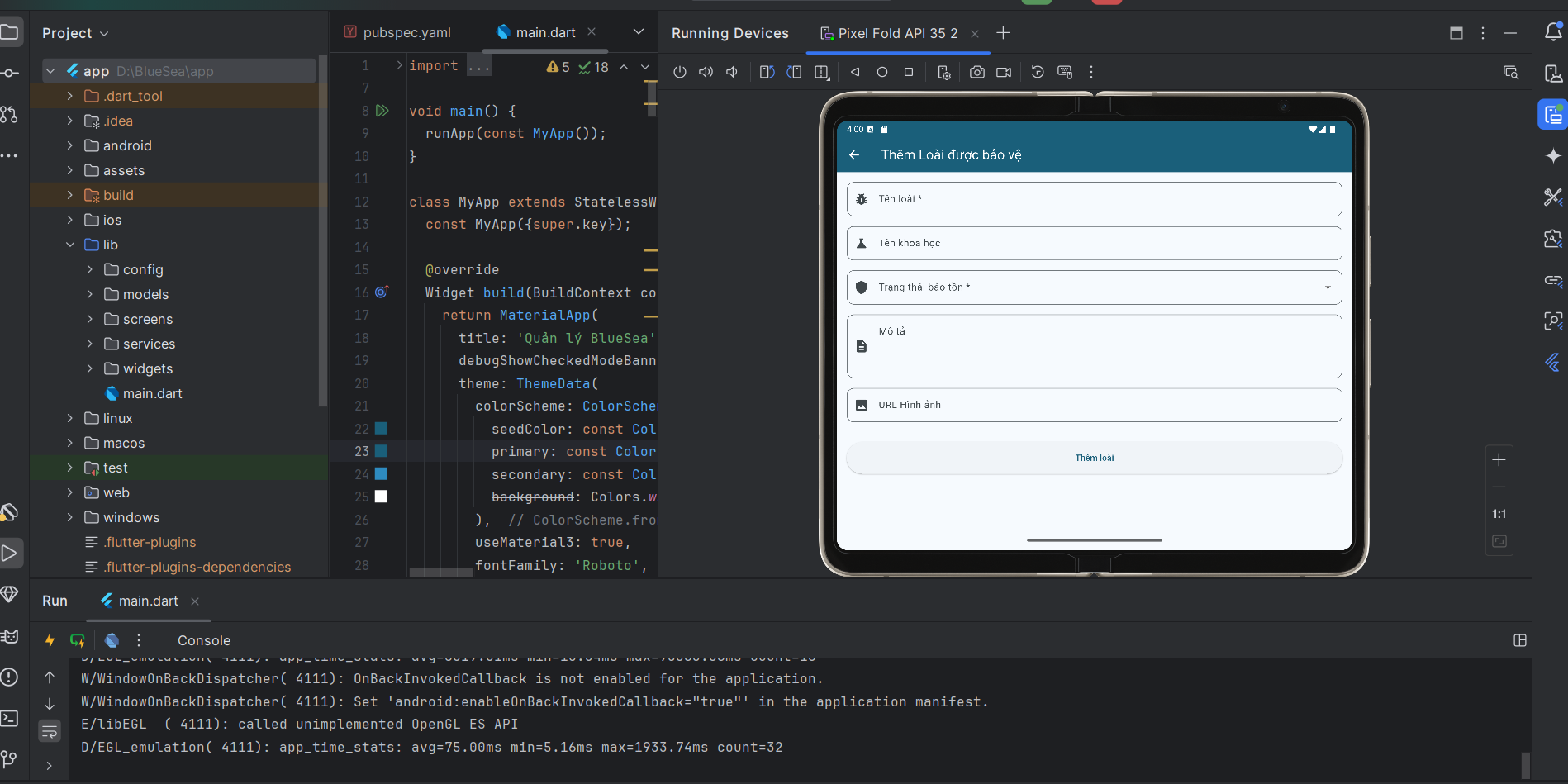
3.8 Giao diện thêm người dùng mới



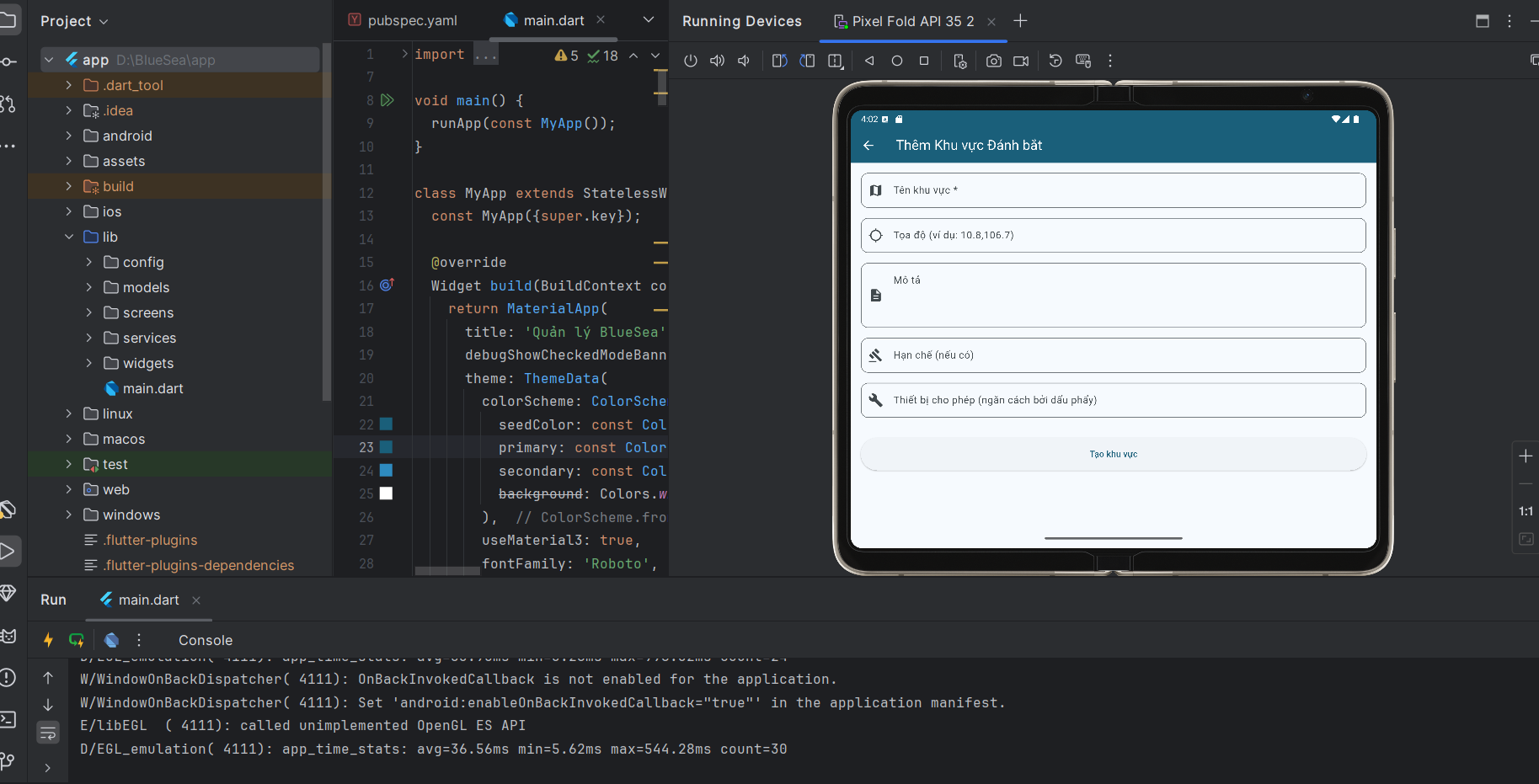
## 3.9 Giao diện edit chỉnh sửa tài nguyên



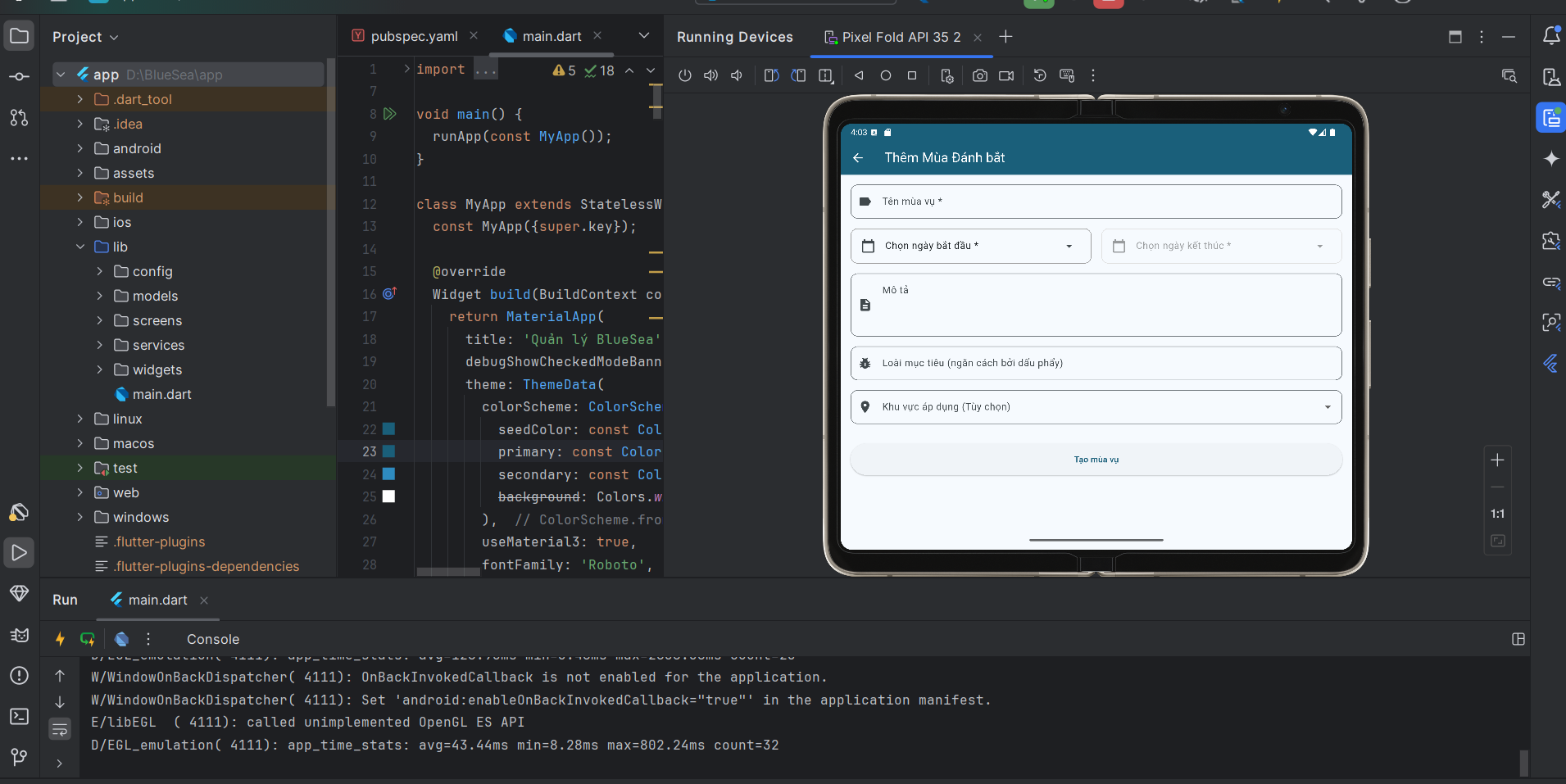
## Giao diện thêm loài cần được bảo vệ



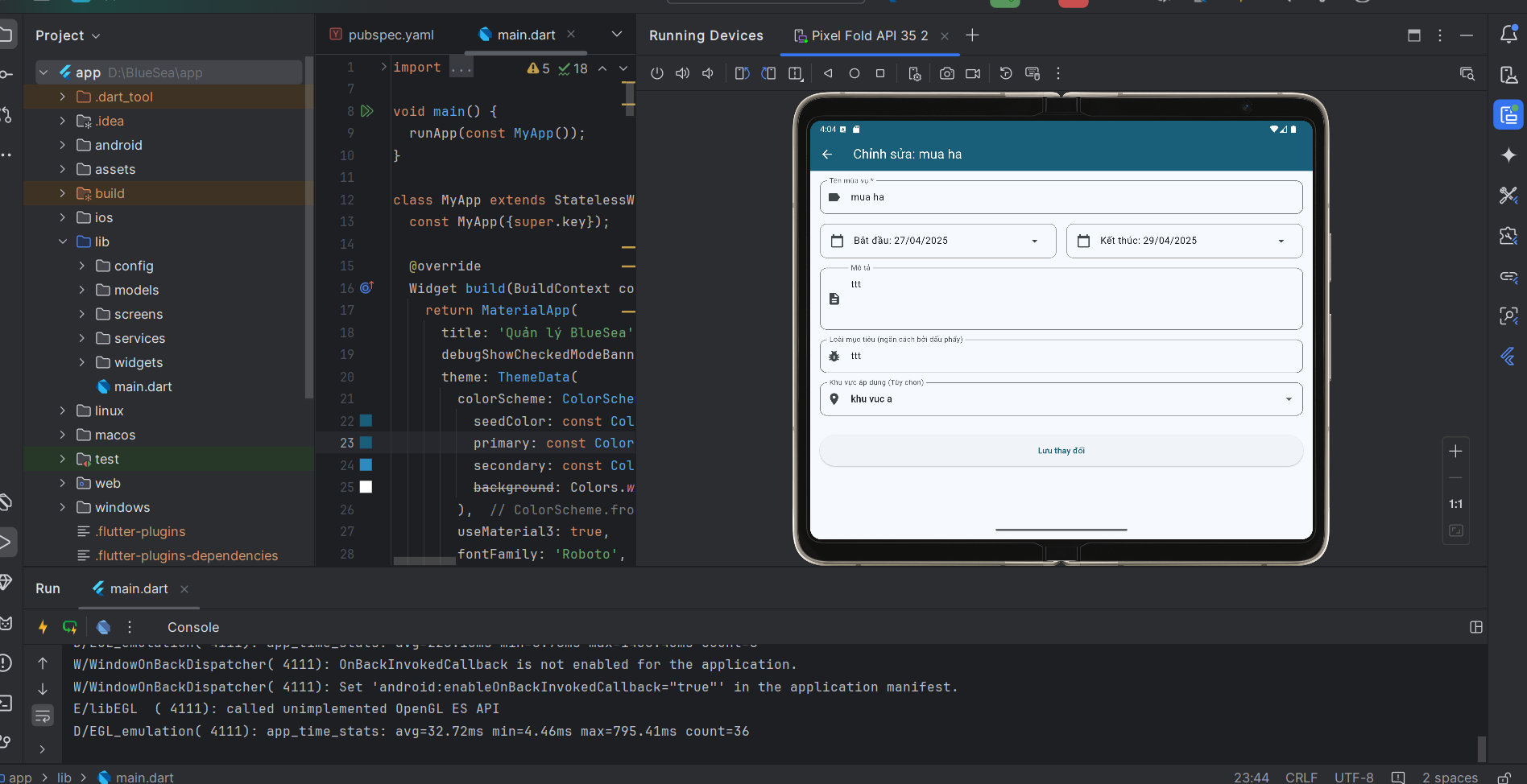
## Giao diện khu vực cần đánh bắt



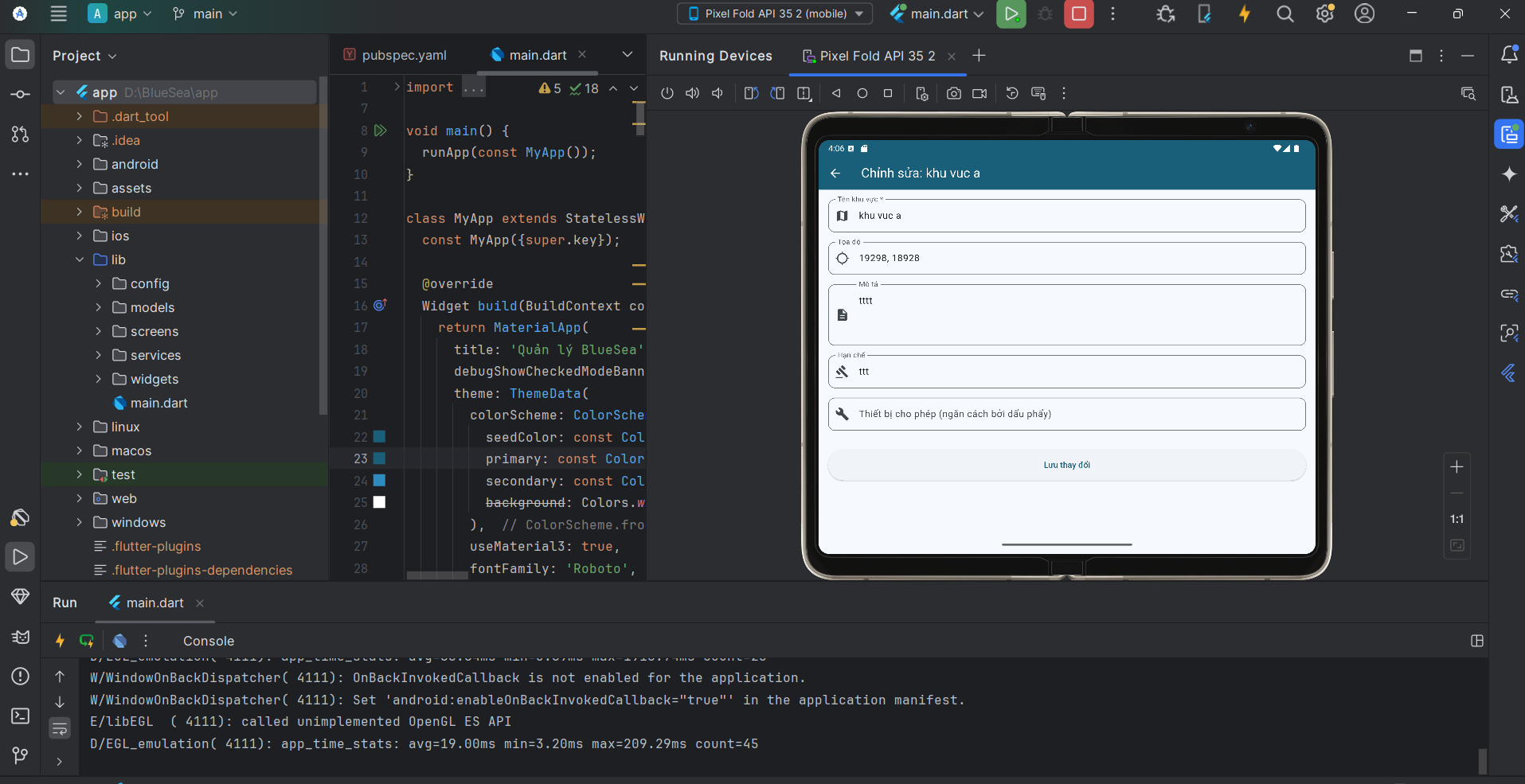
Giao diện thêm mùa được đánh bắt



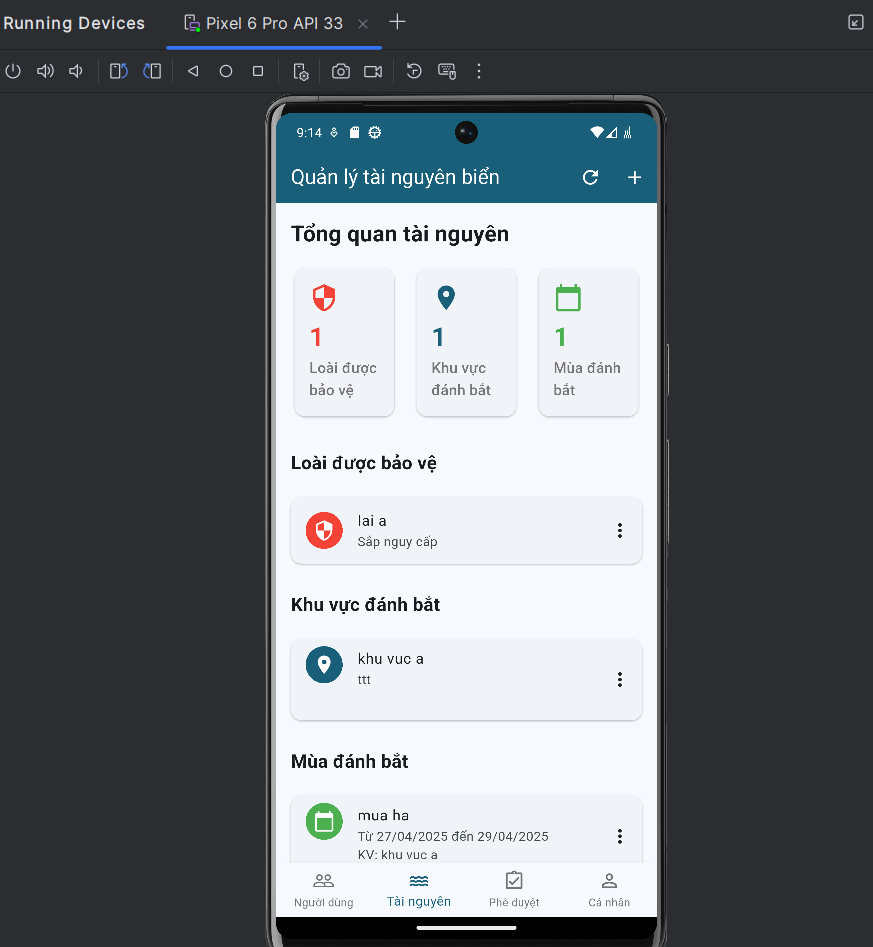
## Giao diện chỉnh sửa mùa được đánh bắt



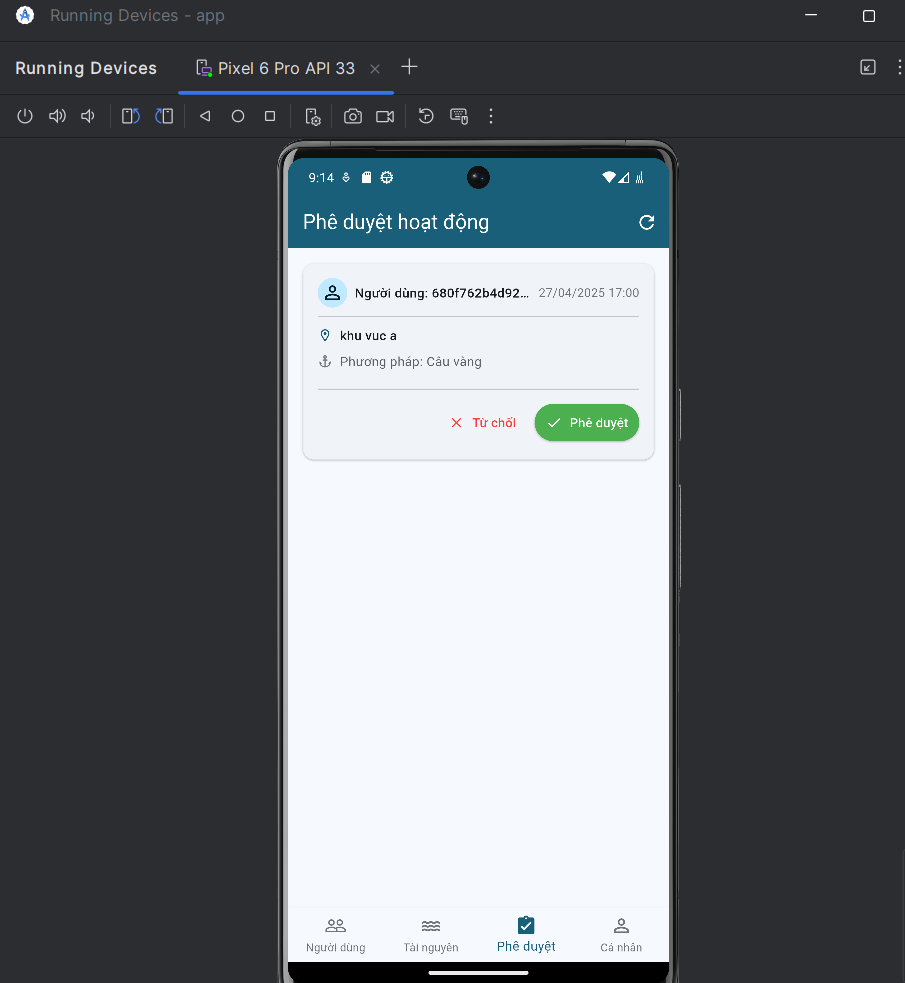
## Giao diện chỉnh sửa khu vực



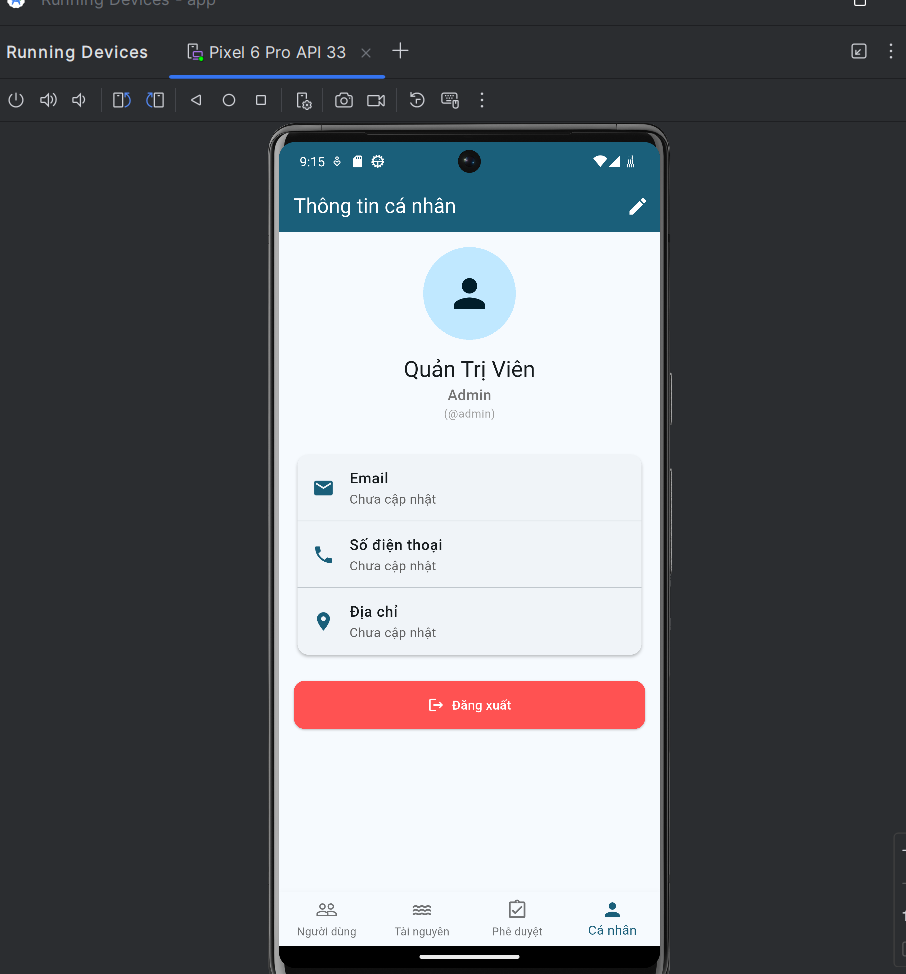
## 3.8Giao diện quản lý tài nguyên



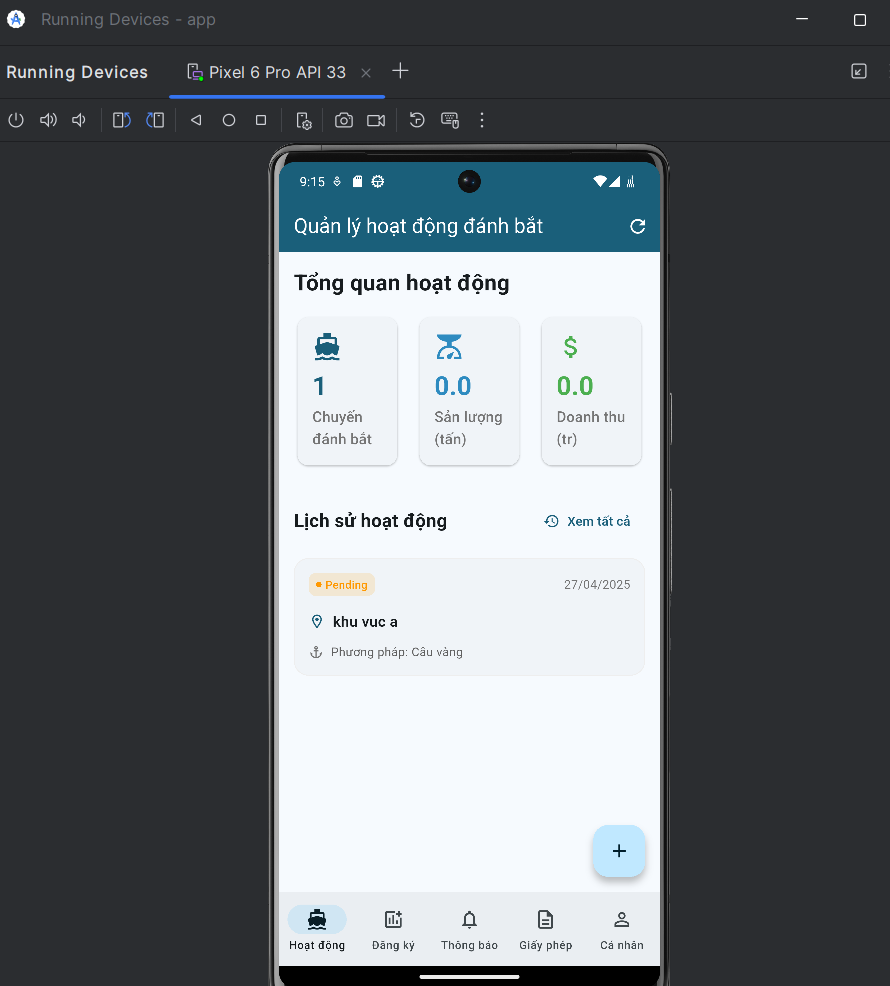
## 3.9Giao diện phê duyệt của admin



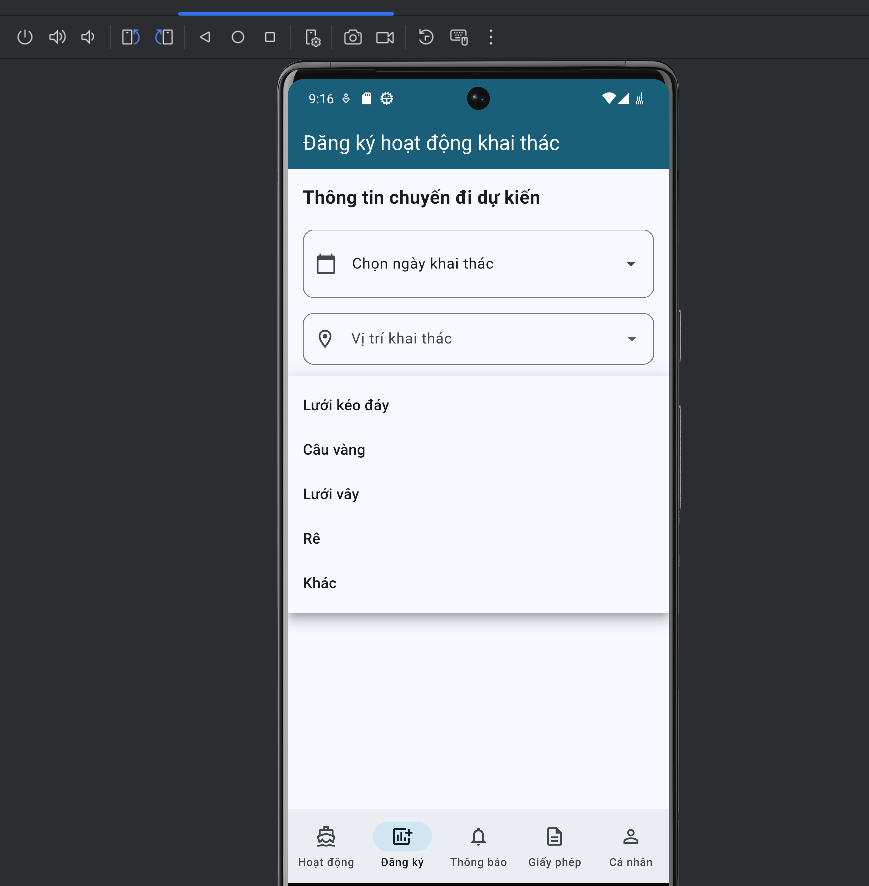
## 3.10Giao diện cá nhân



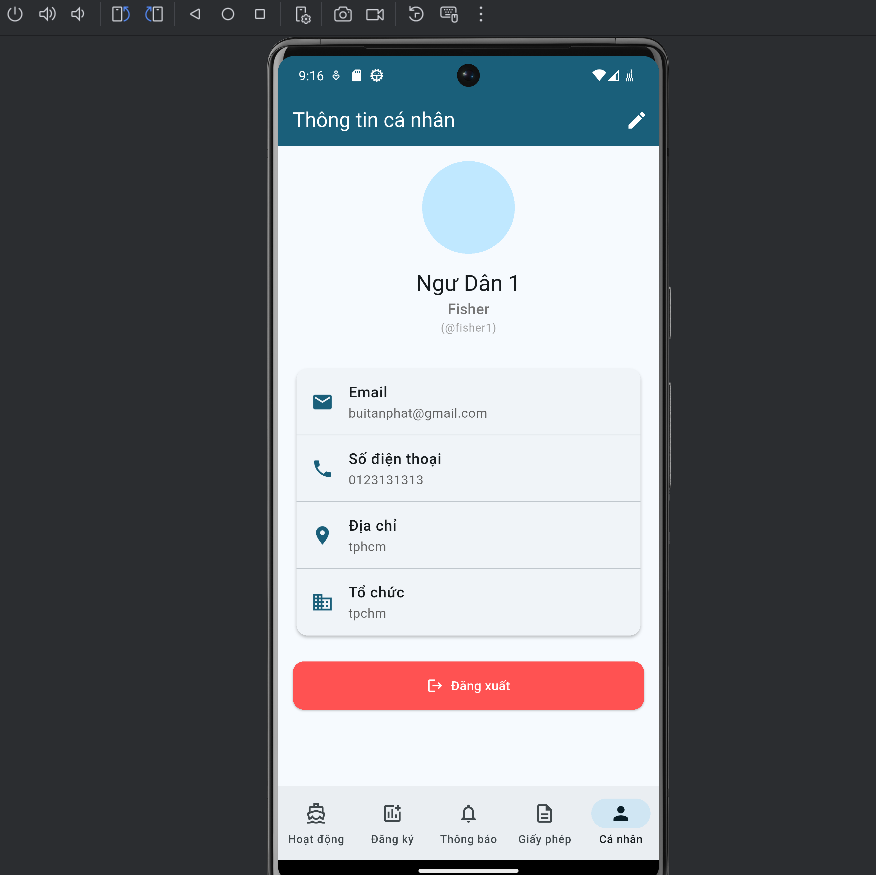
## 3.11Giao diện của ngư dân quản lý



## 3.12Giao diện đăng ký quyền thông qua admin



## 3.13 Thông tin người dùng



# CHƯƠNG 4 KẾT LUẬN

Sau một thời gian nghiên cứu và thực hiện đồ án chuyên ngành với đề tài **“Phát triển phần mềm quản lý thủy sản tại TP.HCM”**, em đã hoàn thiện một hệ thống phần mềm hỗ trợ quản lý toàn diện các hoạt động trong chuỗi sản xuất và phân phối thủy sản, từ khâu nuôi trồng, thu hoạch đến phân phối. Hệ thống được thiết kế nhằm hỗ trợ các hợp tác xã, doanh nghiệp, và cơ quan quản lý có thể dễ dàng theo dõi, thống kê và phân tích dữ liệu một cách hiệu quả, góp phần hiện đại hóa ngành thủy sản tại khu vực TP.HCM.

Trong quá trình thực hiện, em đã lựa chọn kiến trúc công nghệ hiện đại gồm **Node.js** cho backend, **MongoDB** cho cơ sở dữ liệu NoSQL phi quan hệ, kết hợp với **ReactJS** để phát triển giao diện web thân thiện và **Flutter** để xây dựng ứng dụng mobile đa nền tảng. Việc lựa chọn các công nghệ này không chỉ đáp ứng tốt yêu cầu về hiệu suất và khả năng mở rộng, mà còn giúp em học hỏi và làm quen với các công nghệ phổ biến trong các doanh nghiệp phần mềm hiện nay.

Phần mềm được xây dựng để người dùng có thể:

* Đăng ký trại nuôi, nhập và cập nhật thông tin ao hồ, loại thủy sản.
* Ghi nhận thông tin quá trình nuôi (thức ăn, thuốc, điều kiện môi trường).
* Ghi chú sản lượng thu hoạch, nhật ký thu hoạch và phân phối.
* Truy xuất nguồn gốc theo lô sản xuất.
* Quản lý người dùng với hệ thống phân quyền (admin, kỹ thuật viên, người nuôi...).
* Đồng bộ dữ liệu giữa nền tảng web và ứng dụng di động theo thời gian thực qua API.

Việc sử dụng MongoDB giúp hệ thống lưu trữ linh hoạt dữ liệu không đồng nhất, đặc biệt phù hợp với nghiệp vụ thay đổi tùy theo từng mô hình nuôi (cá, tôm, lươn...). Ngoài ra, các API RESTful được xây dựng bằng Node.js đã hỗ trợ tốt việc truyền – nhận dữ liệu giữa frontend và backend, đồng thời đảm bảo bảo mật qua cơ chế xác thực JWT và mã hóa mật khẩu bằng thư viện bcrypt.

Một điểm đáng chú ý là việc ứng dụng **ReactJS** giúp tạo giao diện web tương tác, linh hoạt, phù hợp với người quản lý và cán bộ kỹ thuật; trong khi đó, **Flutter** mang lại trải nghiệm mượt mà cho người nuôi trồng thủy sản có thể ghi nhận dữ liệu ngay tại ao nuôi qua điện thoại di động. Hệ thống hỗ trợ thông báo cảnh báo sớm nếu môi trường nuôi vượt ngưỡng an toàn, thông qua tích hợp với thiết bị IoT mô phỏng (nếu có điều kiện triển khai thực tế).

Trong quá trình phát triển, em đã học được cách tổ chức mã nguồn sạch, chia module hợp lý theo kiến trúc MVC, viết tài liệu API đầy đủ và test tính ổn định của các endpoint. Một số lỗi như mất kết nối cơ sở dữ liệu MongoDB hoặc sai định dạng JSON đầu vào đã giúp em hiểu rõ hơn về cơ chế xử lý bất đồng bộ trong Node.js và tầm quan trọng của việc validate dữ liệu đầu vào với thư viện như Joi hay Mongoose schema.

Bên cạnh những tính năng đã hoàn thiện, hệ thống vẫn còn một số hạn chế cần cải thiện: giao diện quản trị chưa có dashboard phân tích chuyên sâu, thiếu biểu đồ trực quan hóa dữ liệu; chức năng gợi ý lịch thu hoạch hoặc dự báo sản lượng chưa triển khai; chưa tích hợp hệ thống gửi thông báo SMS hoặc email tự động. Đây là các định hướng phát triển trong tương lai, đặc biệt là ứng dụng trí tuệ nhân tạo (AI) và học máy để tối ưu hóa quy trình nuôi trồng.

Ngoài kiến thức chuyên môn, quá trình làm đồ án còn giúp em rèn luyện kỹ năng làm việc nhóm, phân chia nhiệm vụ rõ ràng (trong giai đoạn thiết kế, frontend, backend...), kỹ năng tự học công nghệ mới, và nhất là kiên nhẫn giải quyết lỗi khi hệ thống hoạt động không đúng như mong đợi. Dù không có cơ hội tham gia trực tiếp vào các doanh nghiệp thủy sản, nhưng thông qua việc khảo sát tài liệu thực tế và tham khảo quy trình vận hành từ các mô hình trại nuôi mẫu, em đã phần nào hiểu rõ những vấn đề mà ngành thủy sản đang gặp phải và cách công nghệ có thể hỗ trợ giải quyết.

Tổng kết lại, đồ án chuyên ngành là một trải nghiệm thiết thực, cho em cái nhìn sâu sắc hơn về quá trình phát triển phần mềm từ phân tích nhu cầu đến triển khai sản phẩm hoàn chỉnh. Em tin rằng với nền tảng công nghệ đã lựa chọn, cùng định hướng phát triển lâu dài, phần mềm quản lý thủy sản này có tiềm năng được ứng dụng thực tế tại các địa phương, góp phần chuyển đổi số ngành thủy sản một cách hiệu quả.

**Bài học rút ra – Kinh nghiệm**

1. **Tầm quan trọng của quản lý số hoá:** Việc áp dụng công nghệ trong quản lý giúp minh bạch, nhanh chóng và dễ dàng truy xuất nguồn gốc, từ đó tăng cường hiệu quả kiểm soát và giám sát.
2. **Cần có sự phối hợp liên ngành:** Hiệu quả của hệ thống quản lý phụ thuộc vào sự liên kết chặt chẽ giữa các cơ quan nhà nước, doanh nghiệp và người dân.
3. **Nâng cao nhận thức cộng đồng:** Sự thành công của hệ thống không chỉ đến từ cơ chế quản lý mà còn từ ý thức chấp hành, tuân thủ quy định và hợp tác của người dân.
4. **Linh hoạt trong chính sách:** Quản lý thủy sản cần được điều chỉnh linh hoạt theo biến động thực tế như biến đổi khí hậu, dịch bệnh, hay xu hướng thị trường.
5. **Đầu tư cho nghiên cứu và phát triển:** Phát triển bền vững ngành thủy sản đòi hỏi sự đầu tư lâu dài vào nghiên cứu khoa học, kỹ thuật nuôi trồng, cũng như chính sách hỗ trợ phù hợp.
6. **TÀI LIỆU THAM KHẢO**

 **"Phần mềm ERP ngành thủy sản nâng tầm quản trị, tối ưu chi phí"**  
Bài viết này trình bày về việc áp dụng phần mềm ERP trong ngành thủy sản để nâng cao hiệu quả quản lý và giảm chi phí.

 **"Top các phần mềm quản lý trang trại NTTS (Phần 1)"**  
Bài viết giới thiệu các phần mềm quản lý trang trại nuôi trồng thủy sản, giúp tối ưu hóa hoạt động và tăng cường hiệu quả sản xuất.

 **"TP.HCM: Xây dựng phần mềm quản lý cơ sở dữ liệu khí tượng, thủy văn"**  
Bài báo đề cập đến việc TP.HCM triển khai phần mềm quản lý dữ liệu khí tượng và thủy văn để hỗ trợ công tác phòng chống thiên tai.

 **"Top các phần mềm quản lý trang trại NTTS (Phần 2)"**  
Phần tiếp theo giới thiệu thêm các phần mềm quản lý trang trại nuôi trồng thủy sản, tập trung vào công nghệ AI và giám sát từ xa.

 **"Phần Mềm Quản Trị Sản Xuất, Giá Thành Thủy Sản ERP SEAFOOD"**  
Bài viết trình bày về giải pháp ERP dành cho ngành thủy sản, giúp quản lý sản xuất và tính giá thành hiệu quả.

 **"Thử nghiệm app eCDT giám sát khai thác trái phép thủy sản"**  
Bài báo giới thiệu về ứng dụng eCDT trong việc giám sát và ngăn chặn khai thác thủy sản trái phép.

 **"Ứng dụng phần mềm quản lý tàu cá, minh bạch chống khai thác IUU"**  
Bài viết đề cập đến việc áp dụng phần mềm quản lý tàu cá nhằm tăng cường minh bạch và chống khai thác bất hợp pháp.

 **"Đẩy mạnh công nghệ số trong giám sát, truy xuất nguồn gốc thủy sản"**  
Bài báo nhấn mạnh vai trò của công nghệ số trong việc giám sát và truy xuất nguồn gốc thủy sản.

 **"Triển khai phần mềm điện tử truy xuất nguồn gốc khai thác thủy sản"**  
Bài viết về việc triển khai phần mềm điện tử để truy xuất nguồn gốc thủy sản khai thác, nhằm đảm bảo tính minh bạch và hiệu quả.

 **"Sản phẩm: Xây dựng phần mềm quản lý cơ sở dữ liệu hoạt động sản xuất giống thủy sản"**  
Bài báo giới thiệu về phần mềm quản lý cơ sở dữ liệu trong hoạt động sản xuất giống thủy sản, nhằm nâng cao hiệu quả quản lý.

**GIẢNG VIÊN HƯỚNG DẪN NGƯỜI LẬP ĐỀ CƯƠNG**

(ký và ghi rõ họ tên)

**TRƯỞNG BỘ MÔN**