

**BÁO CÁO TIẾN ĐỘ**

**WEBSITE QUẢN LÝ KHÁCH SẠN**

**TRIỂN KHAI TRÊN AWS**

Họ và tên: Nguyễn Thị Thuý Hiền

MSSV: 2180600395

Vị trí thực tập: FCJ Trainee

Người hướng dẫn: Văn Hoàng Kha

TP. Hồ Chí Minh, 2025

**LỜI NÓI ĐẦU**

Mục lục

[1. Executive Summary 4](#_Toc203052540)

[2. Problem Statement 5](#_Toc203052541)

[3. Solution Architecture 6](#_Toc203052542)

[4. Technical Implementation 7](#_Toc203052543)

[5. Timeline & Milestones 8](#_Toc203052544)

[6. Budget Estimation 10](#_Toc203052545)

[7. Risk Assessment 11](#_Toc203052546)

[7.1 Rủi ro kỹ thuật (Technical Risks) 12](#_Toc203052547)

[7.2 Rủi ro bảo mật (Security Risks) 12](#_Toc203052548)

[7.3 Rủi ro về dữ liệu (Data Risks) 13](#_Toc203052549)

[7.4 Rủi ro triển khai (Deployment & Resource Risks) 13](#_Toc203052550)

[8. Expected Outcomes 14](#_Toc203052551)

# Executive Summary

Trong bối cảnh ngành dịch vụ lưu trú đang phát triển nhanh và cạnh tranh ngày càng khốc liệt, các cơ sở khách sạn vừa và nhỏ (SME) đang gặp nhiều thách thức trong việc quản lý hoạt động vận hành một cách hiệu quả và hiện đại. Phần lớn các khách sạn này vẫn vận hành theo mô hình truyền thống, sử dụng bảng tính Excel hoặc phần mềm cục bộ không có khả năng mở rộng, không hỗ trợ truy cập từ xa, thiếu tích hợp báo cáo và đặc biệt không đảm bảo an toàn bảo mật dữ liệu. Điều này không chỉ ảnh hưởng đến trải nghiệm khách hàng mà còn làm giảm năng suất và khả năng cạnh tranh của đơn vị trên thị trường, nhất là trước làn sóng chuyển đổi số đang diễn ra mạnh mẽ.

Dự án “Hệ thống quản lý khách sạn sử dụng dịch vụ AWS” được thiết kế như một giải pháp toàn diện nhằm giải quyết các bất cập nêu trên. Mục tiêu của dự án là phát triển một nền tảng phần mềm dựa trên công nghệ cloud-native, có thể triển khai nhanh chóng, mở rộng dễ dàng, và tích hợp sâu với hệ sinh thái dịch vụ AWS – từ xác thực người dùng, lưu trữ ảnh, cơ sở dữ liệu, đến phân phối nội dung toàn cầu. Toàn bộ hệ thống được xây dựng với frontend ReactJS, backend NodeJS chạy serverless trên AWS Lambda, dữ liệu được lưu trữ trên Amazon DynamoDB, xác thực thông qua Amazon Cognito và ảnh phòng được lưu trên Amazon S3 và phân phối qua CloudFront

* **Key Functionalities**
* Xác thực người dùng: Đăng ký, đăng nhập, reset mật khẩu, OTP email.
* Quản lý hệ thống phòng – dịch vụ: Tạo, sửa, xóa phòng, upload ảnh.
* Quản lý đặt và trả phòng, thống kê theo thời gian, trạng thái.
* Thanh toán nội bộ (có thể mở rộng tích hợp Momo/VNPay).
* Báo cáo doanh thu: theo tháng, loại phòng, trạng thái khách hàng.
* Giao diện responsive, dễ dùng và dễ tùy chỉnh theo thương hiệu
* **Business Benefits & ROI**
* Tối ưu hóa quy trình vận hành: giảm đến 50% công việc thủ công
* Truy cập mọi lúc, mọi nơi, không cần phần mềm cài đặt.
* Tăng khả năng giám sát và ra quyết định theo thời gian thực
* Khả năng mở rộng sang nhiều điểm lưu trú/khách sạn
* Có thể thương mại hóa dưới dạng SaaS (Software-as-a-Service) cho các khách sạn nhỏ
* **Investment & Success Metrics**
* Hạ tầng AWS chi phí rất thấp nhờ vào mô hình pay-as-you-go
* Chi phí triển khai ban đầu gần như bằng 0 nếu tận dụng đội ngũ sinh viên hoặc kỹ sư nội bộ
* Break-even sau khoảng 4–6 tháng nếu triển khai thực tế
* Tỷ lệ xác thực OTP thành công ≥ 98%, thời gian phản hồi < 500ms, uptime ≥ 99.5%.

# 2. Problem Statement

Hiện nay, phần lớn các khách sạn nhỏ và vừa tại Việt Nam vẫn đang vận hành dựa trên những công cụ truyền thống, điển hình như bảng tính Excel, phần mềm nội bộ được phát triển không đồng nhất, hoặc thậm chí xử lý thủ công bằng giấy tờ. Những giải pháp này thường được triển khai tạm thời trong giai đoạn ban đầu nhằm tiết kiệm chi phí, tuy nhiên về lâu dài lại bộc lộ nhiều hạn chế nghiêm trọng về mặt hiệu quả và độ tin cậy.

Hệ thống quản lý hiện tại của nhiều khách sạn không được thiết kế để xử lý dữ liệu theo thời gian thực, dẫn đến tình trạng quá tải thông tin, khó kiểm tra trạng thái phòng, và đặc biệt là thiếu khả năng truy cập từ xa – một nhu cầu ngày càng cấp thiết trong môi trường quản lý phân tán hoặc đa điểm. Ngoài ra, việc thiếu cơ chế xác thực an toàn, không có hệ thống phân quyền rõ ràng, và không hỗ trợ backup hoặc khôi phục dữ liệu tự động khiến các đơn vị vận hành rơi vào trạng thái dễ bị tổn thương trước các sự cố kỹ thuật hoặc thao tác sai sót từ người dùng.

Việc không có báo cáo phân tích hoạt động cũng khiến các nhà quản lý khách sạn gặp khó khăn trong việc đánh giá hiệu suất khai thác phòng, theo dõi dòng tiền, và đưa ra quyết định kịp thời liên quan đến giá cả, khuyến mãi, hay chiến lược tăng trưởng. Trong bối cảnh ngành dịch vụ lưu trú đang chịu sức ép từ cả thị trường nội địa lẫn các nền tảng đặt phòng trực tuyến (OTA), sự lạc hậu về công nghệ đang trở thành một rào cản lớn ngăn cản các SME trong ngành này phát triển bền vững và mở rộng quy mô kinh doanh.

# 3. Solution Architecture

Giải pháp được thiết kế theo kiến trúc microservices và cloud-native, tận dụng toàn bộ hệ sinh thái AWS để đảm bảo khả năng mở rộng, bảo mật và hiệu suất cao. Hệ thống frontend sử dụng ReactJS để xây dựng SPA (Single Page Application) với Redux Saga giúp quản lý luồng xử lý bất đồng bộ, tích hợp React Router để chuyển trang mượt mà. Giao diện được xây dựng bằng Chakra UI để đảm bảo tính thẩm mỹ và tương thích thiết bị.

Backend sử dụng NodeJS và triển khai dưới dạng Serverless Function thông qua AWS Lambda. Dữ liệu người dùng, phòng, đơn đặt phòng và dịch vụ được lưu trữ trên DynamoDB – cơ sở dữ liệu NoSQL phân tán của AWS. Ảnh phòng được tải lên và lưu trữ trên Amazon S3, sau đó được phân phối nhanh qua Amazon CloudFront để giảm độ trễ và cải thiện trải nghiệm người dùng.

Xác thực và phân quyền người dùng được đảm nhiệm bởi Amazon Cognito, hỗ trợ đa hình thức đăng nhập, quản lý phiên, OTP qua email, reset mật khẩu. Hệ thống sử dụng CloudWatch để ghi nhận log, theo dõi hiệu suất và cảnh báo lỗi.

* Kiến trúc đảm bảo
* Hiệu năng cao nhờ serverless auto-scaling
* Độ tin cậy với S3, DynamoDB, Lambda đều có SLA cao
* Bảo mật với Cognito, phân quyền theo vai trò
* Khả năng tích hợp tương lai với API đặt phòng bên ngoài (OTA)

# 4. Technical Implementation

Triển khai dự án được chia thành 4 giai đoạn chính:

* Giai đoạn 1: Thiết kế và dựng kiến trúc
* Xác định các dịch vụ AWS cần sử dụng, tạo DynamoDB tables, cấu hình Cognito User Pool
* Giai đoạn 2: Phát triển frontend
* Xây dựng layout giao diện, các trang chính (đăng nhập, dashboard, quản lý phòng)
* Tích hợp Redux Saga và xác thực với Cognito
* Giai đoạn 3: Phát triển backend và xử lý logic nghiệp vụ
* Xây dựng các Lambda function xử lý tạo/sửa/xóa phòng, booking, khách hàng
* Tích hợp S3 cho upload ảnh phòng, kết nối DynamoDB bằng AWS SDK
* Giai đoạn 4: Kiểm thử và triển khai
* Unit test cho từng chức năng
* Integration test giữa frontend và backend
* Đưa frontend lên S3 và cấu hình CloudFront
* Triển khai CI/CD với GitHub và CodePipeline
* Chiến lược rollback được thực hiện bằng cách giữ lại các bản release trước trong CloudFront và S3.

# 5. Timeline & Milestones

Dự án được triển khai theo mô hình tuần tự (sequential) kết hợp với mô hình phát triển linh hoạt (agile lightweight) nhằm đảm bảo kiểm soát tiến độ và cho phép điều chỉnh linh hoạt theo phản hồi thực tế trong quá trình phát triển. Tổng thời gian dự kiến triển khai là **6 tuần**, chia làm 6 giai đoạn như sau:

* **Tuần 1 – Phân tích yêu cầu và thiết kế kiến trúc hệ thống:**
  + Thu thập yêu cầu chức năng và phi chức năng từ phía người dùng cuối (chủ khách sạn, nhân viên, khách đặt phòng).
  + Thiết kế kiến trúc hệ thống tổng thể, sơ đồ dữ liệu, luồng nghiệp vụ chính.
  + Lập kế hoạch phân chia công việc

A screenshot of a computer

AI-generated content may be incorrect.

* + **Tuần 2–3 – Phát triển giao diện người dùng và xác thực:**
  + Xây dựng các màn hình đăng nhập, đăng ký, dashboard, quản lý phòng và đặt phòng bằng ReactJS.
  + Tích hợp hệ thống xác thực người dùng với AWS Cognito (xác minh OTP qua email, đổi mật khẩu, phân quyền).
  + A screenshot of a computer

    AI-generated content may be incorrect.Kiểm thử nội bộ giao diện.
  + **Tuần 4 – Phát triển backend và tích hợp dịch vụ AWS:**
  + Xây dựng Lambda function xử lý tạo/xóa/cập nhật phòng, khách hàng, đơn đặt phòng.
  + Kết nối DynamoDB, xử lý luồng dữ liệu giữa frontend và backend.
  + Tích hợp tính năng upload ảnh lên AWS S3, phân phối qua CloudFront.
  + Tạo log và cảnh báo với AWS CloudWatch.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + **Tuần 5 – Kiểm thử toàn hệ thống và chuẩn bị triển khai:**
  + Thực hiện unit test, integration test và performance test.
  + Rà soát bảo mật xác thực người dùng và phân quyền hệ thống.
  + Đánh giá trải nghiệm người dùng trên đa thiết bị (desktop, tablet, mobile).
* **Tuần 6 – Triển khai lên môi trường thật và nghiệm thu:**
  + Đưa giao diện frontend lên AWS S3.
  + Cấu hình CloudFront để phân phối toàn cầu.
  + Tối ưu hóa hiệu suất, thực hiện backup cấu hình.
  + Đánh giá nghiệm thu, ghi nhận lesson learned.

**Các mốc quan trọng (Milestones):**

* Tuần 2: Hoàn tất giao diện người dùng chính.
* Tuần 3: Hoàn thành tích hợp Cognito và logic xác thực.
* Tuần 4: Chạy bản demo với các chức năng cơ bản (CRUD, booking, upload ảnh).
* Tuần 6: Triển khai chính thức lên AWS, nghiệm thu nội bộ

# 6. Budget Estimation

Dự án áp dụng kiến trúc serverless và mô hình tính phí theo tài nguyên tiêu thụ (pay-as-you-go), do đó chi phí duy trì hạ tầng cực kỳ thấp, đặc biệt phù hợp với các dự án học thuật hoặc khởi nghiệp giai đoạn đầu. Việc tận dụng AWS Free Tier cho phép tiết kiệm tối đa trong giai đoạn phát triển và thử nghiệm.

Chi phí dịch vụ AWS (ước tính tối thiểu):

* Amazon S3 (Lưu ảnh): 5–10 GB/tháng dùng để lưu ảnh phòng khách sạn. Với mức truy cập thấp, chi phí duy trì dao động khoảng 0.2–0.5 USD/tháng.
* Amazon CloudFront (Phân phối ảnh): Với khoảng 50GB truyền tải/tháng, chi phí khoảng 4–5 USD/tháng, nhưng có thể thấp hơn nếu chưa đến ngưỡng miễn phí.
* AWS Cognito (Xác thực người dùng): Miễn phí cho 50.000 MAUs đầu tiên/tháng, đủ cho toàn bộ giai đoạn thử nghiệm và vận hành SME.
* Amazon DynamoDB (Cơ sở dữ liệu): Free Tier hỗ trợ 25GB lưu trữ, 25 RCU + 25 WCU. Nếu vượt mức sẽ tính chi phí thêm dựa trên dung lượng và throughput (vẫn rất rẻ so với RDS hoặc EC2).
* AWS Lambda (Serverless backend): Miễn phí 1 triệu request đầu tiên mỗi tháng. Với ứng dụng SME, phần lớn hoạt động nằm trong free tier.

Chi phí phát triển:

* Nhân sự: Miễn phí nếu sử dụng nguồn lực sinh viên, hoặc tài trợ từ chương trình học thuật.
* Công cụ phụ trợ: Toàn bộ công cụ phát triển là mã nguồn mở (ReactJS, NodeJS), không tốn license.

Tổng chi phí duy trì ước tính: < 10 USD/tháng phù hợp với nhóm khách sạn SME không có ngân sách CNTT lớn.

ROI (Return on Investment):

* Hoàn vốn sau 4–6 tháng nếu triển khai thực tế cho 1 cơ sở có 20–50 phòng nhờ:
  + Giảm chi phí vận hành, sai sót trong booking.
  + Tăng hiệu suất làm việc và tự động hóa quy trình.
  + Tăng trải nghiệm và khả năng giữ chân khách hàng.

# 7. Risk Assessment

Việc triển khai một hệ thống dựa hoàn toàn trên các dịch vụ cloud-native như AWS đòi hỏi đánh giá cẩn trọng các rủi ro tiềm ẩn để đảm bảo tính ổn định, bảo mật và khả năng mở rộng của hệ thống. Dưới đây là các nhóm rủi ro chính được xác định trong phạm vi dự án và chiến lược ứng phó tương ứng

## **7.1 Rủi ro kỹ thuật (Technical Risks)**

Một số thành viên trong nhóm phát triển chưa có nhiều kinh nghiệm với kiến trúc serverless và mô hình tích hợp các dịch vụ phi máy chủ như Lambda, DynamoDB, S3. Điều này có thể dẫn đến lỗi triển khai do chưa nắm rõ cách hoạt động bất đồng bộ, giới hạn tài nguyên, hoặc cách xử lý sự kiện theo chuỗi.

Giải pháp ứng phó: Triển khai phương pháp phát triển và kiểm thử theo từng đơn vị nhỏ (micro-testing). Cụ thể, từng Lambda function và luồng xử lý sẽ được xây dựng, kiểm thử và xác minh riêng lẻ trước khi tích hợp vào hệ thống tổng thể. Ngoài ra, sử dụng AWS SAM hoặc Serverless Framework để mô phỏng môi trường triển khai trước khi deploy chính thức.

## 7.2 Rủi ro bảo mật (Security Risks)

Việc thiết lập phân quyền trong AWS Cognito có thể gặp sai sót nếu không xác định rõ vai trò người dùng hoặc bỏ sót cấu hình nhóm truy cập (user groups). Điều này dẫn đến khả năng người dùng truy cập vào các chức năng không phù hợp, gây rò rỉ dữ liệu hoặc lỗi logic nghiệp vụ.

Giải pháp ứng phó: Thiết lập vai trò người dùng rõ ràng từ giai đoạn thiết kế (admin, nhân viên, khách đặt phòng), sử dụng Cognito User Groups để phân tách quyền hạn và kiểm soát truy cập chi tiết. Đồng thời, tiến hành kiểm thử bảo mật xuyên suốt quá trình phát triển.

## 7.3 Rủi ro về dữ liệu (Data Risks)

DynamoDB là cơ sở dữ liệu NoSQL có tốc độ cao nhưng dễ bị lỗi mất mát dữ liệu nếu không cấu hình backup hoặc TTL (Time To Live) hợp lý, nhất là trong các tình huống update hoặc delete nhầm lẫn.

Giải pháp ứng phó: Kích hoạt chế độ Continuous Backup & Point-in-time Recovery trong DynamoDB để tự động lưu trạng thái hệ thống mỗi ngày. Thiết lập chính sách TTL cho các bảng không cần lưu trữ lâu dài. Thực hiện snapshot định kỳ theo tháng, lưu trữ backup vào S3 để đảm bảo an toàn lâu dài.

## 7.4 Rủi ro triển khai (Deployment & Resource Risks)

Do nhóm triển khai phần lớn là sinh viên hoặc nhóm học thuật, có khả năng một số thành viên chưa quen thao tác với AWS Console, CLI hoặc quy trình triển khai CI/CD

Giải pháp ứng phó: Tổ chức buổi hướng dẫn nội bộ ngay từ đầu dự án, phân công vai trò rõ ràng dựa trên mức độ thành thạo kỹ thuật. Áp dụng chính sách mentor nội bộ và ghi lại checklist thao tác triển khai giúp chuẩn hóa quy trình và hỗ trợ thành viên mới

# 8. Expected Outcomes

Việc triển khai hệ thống quản lý khách sạn tích hợp các dịch vụ AWS hiện đại được kỳ vọng sẽ mang lại một loạt kết quả rõ ràng và có thể đo lường được cả về mặt kỹ thuật, vận hành lẫn hiệu quả kinh doanh. Các kết quả mong đợi được chia thành ba giai đoạn theo thời gian như sau:

* Kết quả kỹ thuật và vận hành (tức thời)
* Hệ thống duy trì mức độ hoạt động ổn định (uptime) ≥ 99.5%, bảo đảm sẵn sàng phục vụ người dùng mọi lúc, kể cả trong giờ cao điểm hoặc khi có lưu lượng truy cập tăng đột biến.
* Người dùng cuối có thể đặt phòng, nhận mã OTP xác thực qua email trong vòng dưới 5 giây, nhờ vào tích hợp AWS Cognito kết hợp với SES/SMTP gateway.
* Giao diện web được xây dựng với ReactJS SPA và Chakra UI, đảm bảo trải nghiệm liền mạch trên cả thiết bị desktop và di động, không cần tải lại trang, tăng độ hài lòng cho người sử dụng.
* Chủ khách sạn có thể truy cập hệ thống dashboard phân tích và báo cáo từ xa thông qua giao diện web được bảo vệ bằng xác thực đa tầng.
* Quy trình vận hành như đặt phòng, kiểm tra phòng trống, quản lý thanh toán và khách hàng được tự động hóa, giúp rút ngắn 30–50% thời gian xử lý nghiệp vụ so với quy trình thủ công truyền thống.
* Lợi ích ngắn hạn (0–6 tháng)
* Hoàn vốn nhanh thông qua việc cắt giảm chi phí nhân sự quản lý, giảm sai sót do thao tác tay và loại bỏ nhu cầu vận hành nhiều phần mềm rời rạc không tương thích
* Tăng tốc độ phục vụ và phản hồi khách hàng, từ đó nâng cao trải nghiệm người dùng, tăng tỷ lệ giữ chân khách hàng cũ và tỷ lệ chuyển đổi đặt phòng qua kênh online.
* Hệ thống có thể được sử dụng ngay sau 6 tuần triển khai, không yêu cầu đầu tư phần cứng, giúp khách sạn tiết kiệm thời gian và chi phí triển khai ban đầu.
* Lợi ích trung hạn (6–18 tháng)
* Tích hợp được với các nền tảng OTA (Online Travel Agencies) như Agoda, Booking.com hoặc đối tác trung gian API khác, mở rộng kênh phân phối phòng và quản lý đồng bộ lịch phòng.
* Khả năng mở rộng hệ thống cho nhiều chi nhánh trong cùng chuỗi khách sạn, giúp ban quản lý theo dõi tập trung, đồng nhất quy trình, báo cáo và hiệu quả vận hành toàn hệ thống.
* Phát triển thêm các chức năng phụ trợ như quản lý dịch vụ, voucher khuyến mãi, thanh toán online để nâng cao mức độ cạnh tranh.
* Lợi ích dài hạn (>18 tháng)
* Xây dựng mô hình SaaS (Software-as-a-Service) phục vụ thị trường khách sạn SME, mở rộng cung cấp cho nhiều đối tượng khách hàng mới, từ đó tạo nguồn thu bền vững và mở rộng quy mô hoạt động.
* Hệ thống dần tích lũy dữ liệu vận hành, giao dịch và hành vi người dùng, cho phép áp dụng các công cụ phân tích nâng cao như Business Intelligence (BI) và AI để hỗ trợ quyết định chiến lược, tối ưu hóa doanh thu và phân khúc khách hàng.
* Hướng tới nền tảng cloud-native hoàn toàn, dễ bảo trì, dễ cập nhật, tiết kiệm chi phí hạ tầng, sẵn sàng tích hợp với các công cụ marketing và quản lý khách hàng nâng cao trong tương lai.