

KẾ HOẠCH TRIỂN KHAI DỰ ÁN: URL SHORTENER MICROSERVICES

Mục tiêu: Distinction Level (9 - 10 Points) **Mô hình:** Centralized Architecture (Leader-Driven) **Thời gian:** 4 Tuần



LƯU Ý QUAN TRỌNG

Về Sĩ số Nhóm: Kế hoạch này áp dụng cho nhóm 4 thành viên với giả định đã được giảng viên chấp thuận ngoại lệ. Nếu chưa, vui lòng xác nhận lại ngay.

II. PHÂN CHIA VAI TRÒ (RACI MATRIX)

1. DucAnh (Leader) - *System Architect & Core*

- **Trọng trách:** Chịu trách nhiệm 100% về kiến trúc hệ thống và các phần khó nhất.
- **Nhiệm vụ chính:**
 - Thiết kế kiến trúc Microservices, Database Schema.
 - Code **API Gateway** (Ocelot).
 - Code **Shortener Service** (Logic cốt lõi, CQRS).
 - Cấu hình **RabbitMQ** và **Redis**.
 - Review và Merge code của cả nhóm.

2. Đ.A (Backend Dev) - *Identity & Security*

- **Nhiệm vụ chính:**
 - Phát triển **Users Service** (Module Authentication).

- Xử lý: Đăng ký, Đăng nhập, JWT Token generation/validation.
- Viết Unit Tests cho phần Auth.
- Phụ trách DevOps cho service của mình (Dockerfile, CI Pipeline).

3. Tiến (Backend Dev) - *Management & Data*

- **Nhiệm vụ chính:**

- Phát triển **Users Service** (Module Management).
- Xử lý CRUD: Quản lý Profile, Danh sách Link cá nhân (My Links), Xóa Link.
- Viết Unit Tests cho phần CRUD.
- Phụ trách DevOps cho service của mình.

4. Tuần (Frontend & QA) - *UI/UX & Integration*

- **Nhiệm vụ chính:**

- Xây dựng **Web App** (React/Vue).
- Tích hợp API từ Gateway vào giao diện.
- Viết **Integration Tests** (Kiểm thử luồng chạy từ UI -> Server).
- Soạn thảo tài liệu và Slide thuyết trình.



II. LỘ TRÌNH TRIỂN KHAI CHI TIẾT



TUẦN 1: KHỞI TẠO & LUẬT CHƠI (FOUNDATION)

Mục tiêu: Leader dựng xong khung sườn, quy định xong API để Đ.A và Tiến có thể bắt tay vào làm ngay.



DucAnh (Leader)

- **Repo Setup:** Khởi tạo Git, chia nhánh main , develop .
- **Infrastructure:** Viết docker-compose.yml chạy SQL, Redis, RabbitMQ.

- **Gateway:** Code xong khung Ocelot Gateway.
- **API Contracts (Quan trọng nhất):**
 - Quy định Input/Output cho API Login/Register (Giao cho **Đ.A**).
 - Quy định Input/Output cho API CRUD Link (Giao cho **Tiến**).
 - Gửi file Swagger giả lập cho **Tuấn**.

Team (**Đ.A, Tiến, Tuấn**)

- Clone Repo, cài đặt môi trường (Docker, .NET, Node.js).
 - Chạy thử hệ thống local.
 - Đọc kỹ API Contract mà Leader giao.
-

TUẦN 2: PHÁT TRIỂN ĐỘC LẬP (CORE FEATURES)

Mục tiêu: Ai làm việc này, code chạy được trên máy cá nhân.

DucAnh (Leader - Hard Mode)

- Code **Shortener Service**:
 - Logic sinh mã rút gọn (Base62).
 - Kết nối SQL Database lưu dữ liệu Link.
 - Setup cấu trúc CQRS.
- Cấu hình Gateway chặn các request không có Token hợp lệ.

Đ.A (Auth)

- Code API POST /register : Tạo user mới, hash password.
- Code API POST /login : Kiểm tra pass -> Trả về JWT Token đúng format Leader yêu cầu.

Tiến (CRUD)

- Code API GET /profile , PUT /profile : Lấy/Sửa thông tin user.

- Code API GET /my-links : Lấy danh sách link của user từ DB.
- Code API DELETE /link/{id} : Xóa link.

Tuân (Frontend)

- Dựng khung ReactJS/VueJS.
 - Code giao diện Login/Register (kết nối API local của Đ.A).
 - Code giao diện Dashboard "My Links" (hiển thị dữ liệu giả hoặc API của Tiến).
-

TUẦN 3: TÍCH HỢP NÂNG CAO (DISTINCTION PHASE)

Mục tiêu: Leader thêm công nghệ xịn (RabbitMQ, Redis). Các thành viên hoàn thiện DevOps.

DucAnh (Leader)

- **Redis (Caching):** Cập nhật logic GET /{code} : Check Redis trước khi gọi DB.
- **RabbitMQ (Async):**
 - Khi tạo link -> Bắn event LinkCreated .
 - (Optional) Viết logic log activity khi nhận event này.
- Hỗ trợ Đ.A và Tiến sửa lỗi kết nối.

Đ.A & Tiến

- Viết **Unit Tests** (xUnit) cho các API đã code (Login, Register, CRUD).
- Viết **Dockerfile** (Multi-stage) cho Service của mình.

Tuân

- Viết **Dockerfile** cho Frontend (Nginx).
 - Đổi endpoint Frontend: Trỏ tất cả về **API Gateway** (thay vì gọi trực tiếp Đ.A hay Tiến).
-

● TUẦN 4: GHÉP NỐI & VỀ ĐÍCH (INTEGRATION & DEPLOY)

Mục tiêu: Hệ thống chạy mượt trên Cloud.

👤 DucAnh (Leader)

- **Merge Code:** Review code của Đ.A, Tiến, Tuấn -> Merge vào main .
- **Gateway Config:** Cấu hình Ocelot trỏ đúng vào Container của User Service và Shortener Service.
- **Cloud Setup:** Tạo DB, Redis, RabbitMQ trên Render/Cloud.

👥 Toàn nhóm (CI/CD)

- Thiết lập **GitHub Actions** riêng cho từng người:
 - **Pipeline 1 (Đ.A & Tiến):** Build & Deploy User Service.
 - **Pipeline 2 (DucAnh):** Build & Deploy Shortener Service + Gateway.
 - **Pipeline 3 (Tuấn):** Build & Deploy Frontend.

👤 Tuấn (QA Focus)

- Thực hiện **End-to-End Test:** Đăng ký -> Login -> Tạo Link -> Check Redis -> Redirect.
 - Hoàn thiện Slide thuyết trình & Báo cáo.
-

✓ CHECKLIST PHÂN CÔNG (ĐỂ ĐẠT ĐIỂM)

Hạng mục	Công nghệ	Người chịu trách nhiệm
Architecture	Microservices, CQRS	DucAnh
Communication	RabbitMQ (Msg Broker)	DucAnh
Performance	Redis (Caching)	DucAnh
Gateway	Ocelot	DucAnh

Hạng mục	Công nghệ	Người chịu trách nhiệm
Auth Service	JWT, Identity	Đ.A
CRUD Logic	EF Core, SQL	Tiến
Frontend	React/Vue, Tailwind	Tuấn
Testing	Unit & Integration Tests	Tuấn (Lead) + All
CI/CD	GitHub Actions (Independ.)	Mỗi người tự làm 1 pipe