Tài Liệu Tổng Quan Ứng Dụng Chấm Công

1. Giới thiệu

Ứng dụng chấm công gồm hai đối tượng:

- **Người dùng (Employee):** đăng nhập, xem công.
- **Quản trị (Admin):** xem công của toàn bộ nhân viên, chỉnh sửa công.

Hệ thống đơn giản nhưng yêu cầu bảo mật cao, sử dụng:

- **Kong API Gateway** làm lớp bảo vệ API.
- **JWT Access Token + Refresh Token**.
- **Role-Based Access Control (RBAC).**
- **API auditing**.

---

2. Chức năng chính

2.1 Nhân viên (Employee)

- Đăng nhập
- Xem bảng công cá nhân
- Xem chi tiết từng ngày (giờ vào, giờ ra, ghi chú)

2.2 Admin

- Xem bảng công của tất cả nhân viên
- Chỉnh sửa công (update giờ vào/ra, thêm ghi chú)
- Xem log chỉnh sửa

---

3. Kiến trúc tổng quát

```
[Mobile/Web App]
    ↓ (Access Token)
[Kong Gateway]
    - JWT Validate
    - Rate Limit
    - RBAC Enforcement
    - Logging
    ↓
[Attendance API Service]
    - Auth
    - Attendance CRUD
    - Role Checking
```

---

4. Thiết kế Database (Đơn giản)

4.1 Bảng users

field	type	note
id	int	PK
username	varchar	unique
password_hash	varchar	bcrypt
role	enum('EMP','ADMIN')	phân quyền

4.2 Bảng attendance

field	type	note
id	int	PK
userId	int	FK users.id
date	date	ngày công
checkIn	datetime	giờ vào
checkOut	datetime	giờ ra
note	text	ghi chú
updatedBy	int	admin chỉnh sửa
updatedAt	datetime	log chỉnh sửa

---

5. API Specification

5.1 Auth API

POST /auth/login

Request:

```
{
  "username": "dat",
  "password": "123456"
}
```

Response:

```
{
  "accessToken": "<jwt>",
  "refreshToken": "<jwt>"
}
```

POST /auth/refresh

Lấy token mới khi hết hạn.

5.2 Attendance API

GET /attendance/me (EMP)

Trả về bảng công của chính mình.

GET /attendance/user/:id (ADMIN)

Xem công của nhân viên khác.

PUT /attendance/:id (ADMIN)

Update giờ vào/ra hoặc ghi chú.
Request sample:

```
{
  "checkIn": "2025-12-01T08:00:00",
  "checkOut": "2025-12-01T17:00:00",
  "note": "Đi họp sớm"
}
```

---

6. Token & Bảo mật API

6.1 Access Token

- Có thời hạn ngắn (15–30 phút)
- Lưu trên memory (không lưu localStorage để tránh XSS)

Payload mẫu:

```
{
  "sub": "123",
  "role": "ADMIN",
  "exp": 1733040000
}
```

6.2 Refresh Token

- Hạn dùng dài hơn (7–30 ngày)
- Lưu ở HttpOnly Cookie hoặc Secure Storage
- Có bảng revoke token để logout

6.3 RBAC (Role-Based Access Control)

Luồng:

1. App gửi request
2. Kong check token
3. Backend kiểm tra `role`
4. Nếu không đủ quyền: trả 403

Pseudo code backend:

```
function requireAdmin(req, res, next) {
  if (req.user.role !== 'ADMIN') return res.status(403).json({ message:
"Forbidden" });
  next();
}
```

---

7. Cấu hình Kong (JWT + Rate Limit)

7.1. Enable JWT Plugin

```
kongctl plugins enable jwt
```

7.2. Add consumer + key

```
curl -X POST http://localhost:8001/consumers/attendance
curl -X POST http://localhost:8001/consumers/attendance/jwt --data
"secret=super_secret_key"
```

7.3. Rate Limit

VD: tối đa 20 requests / phút

```
curl -X POST http://localhost:8001/services/attendance/plugins \
  --data "name=rate-limiting" \
  --data "config.minute=20"
```

8. Luồng đăng nhập & gia hạn token

```
[User] -> /auth/login -> [Backend]
Backend tạo access_token + refresh_token
User gọi API bằng access_token
Nếu 401 (expired): app gọi /auth/refresh
token mới -> tiếp tục
```

9. Giao diện App (mockup)

Màn hình Employee

- Login
- Dashboard xem công
- Chi tiết ngày

Màn hình Admin

- Danh sách nhân viên
- Chọn nhân viên → xem công
- Chỉnh sửa công

10. Todo (mở rộng trong file sau)

- Kong decK file mẫu (infrastructure as code)
- Full API code bằng Node.js/Express
- UI mẫu html