

1. Cấu trúc dự án

1.1. Mục tiêu chương

Chương này quy định cấu trúc dự án chuẩn và phạm vi quản lý mã nguồn giữa Core Team và các đơn vị Partner nhằm:

- Đảm bảo tính ổn định của hệ thống
- Tránh xung đột mã nguồn giữa các đơn vị phát triển
- Chuẩn hóa cách tổ chức code trong toàn dự án
- Dễ dàng bảo trì, mở rộng và tích hợp hệ thống

1.2. Khái niệm / phạm vi áp dụng

Tài liệu áp dụng cho:

- Tất cả các dự án sử dụng cấu trúc chuẩn của hệ thống
- Core Team (đội phát triển lõi)
- Partner Team (đối tác phát triển module)

Cấu trúc dự án chuẩn:

com.example.demo

├─ common/ (thành phần dùng chung)

├─ config/ (cấu hình hệ thống)

├─ modules/ (các module nghiệp vụ)

└─ DemoApplication.java

Ý nghĩa từng khu vực

Thư mục	Vai trò	Phạm vi sử dụng
common	Thành phần dùng chung toàn hệ thống	Core quản lý
config	Cấu hình hệ thống, security, hạ tầng	Core quản lý
modules	Các module nghiệp vụ	Core + Partner

1.3. Quy định chính

Phân quyền quản lý code

Khu vực	Đơn vị quản lý	Quyền chỉnh sửa
common/**	Core Team	Partner không được sửa
config/**	Core Team	Partner không được sửa
modules/**	Core + Partner	Được phép phát triển

Quy định bắt buộc

Đối tác KHÔNG được phép sửa trực tiếp:

common/**

config/**

Mọi thay đổi trong các khu vực này phải:

1. Tạo yêu cầu thay đổi (Core Change Request)
2. Được Core Team xem xét
3. Core Team phê duyệt và thực hiện

1.4. Cách thực hiện / quy trình

Quy trình thay đổi code dùng chung

Bước 1: Partner phát hiện nhu cầu thay đổi

Bước 2: Tạo ticket với tiêu đề: các phần mềm quản lý công việc liên quan (Jira / Redmine / YouTrack) hoặc nội bộ

[CORE CHANGE REQUEST] Thêm field phone vào user_account

Yêu cầu: [CORE CHANGE REQUEST] Thêm field phone vào user_account

Mô tả yêu cầu

1. Thông tin người yêu cầu

- Đơn vị:
- Người phụ trách:
- Module liên quan:

2. Mô tả thay đổi

Mô tả chi tiết thay đổi cần thực hiện.

Ví dụ:

Cần thêm field phone vào bảng user_account để phục vụ xác thực OTP.

3. Phạm vi ảnh hưởng

- File/Entity:
- Bảng DB:
- API liên quan:
- Module bị ảnh hưởng:

4. Lý do thay đổi

Mô tả lý do nghiệp vụ hoặc kỹ thuật.

5. Phương án đề xuất

Ví dụ:

- Thêm field: phone VARCHAR(20)

Hạng mục: Quản lý hợp đồng khách hàng

Số hiệu: Cao

Số ưu tiên: Cao

Người giao: Phùng Mạnh Tuấn

Người xử lý: Tìm kiếm

Ngày giao: 09/02/2026

Thời hạn: 09/02/2026

Ngày hoàn thành: 09/02/2026

Trạng thái: Công việc mới

Luồng xử lý:

1. Partner tạo ticket
2. Core Team review
3. Chuyển trạng thái:

Trạng thái	Ý nghĩa
Open	Ticket mới
Under Review	Core đang đánh giá
Approved	Đồng ý thay đổi
Rejected	Từ chối
Implemented	Core đã code
Merged	Đã merge vào develop

Bước 3: Core Team đánh giá:

- Phạm vi ảnh hưởng
- Rủi ro hệ thống

- Tính cần thiết

Bước 4: Nếu hợp lệ:

- Core Team thực hiện thay đổi
- Merge vào nhánh develop

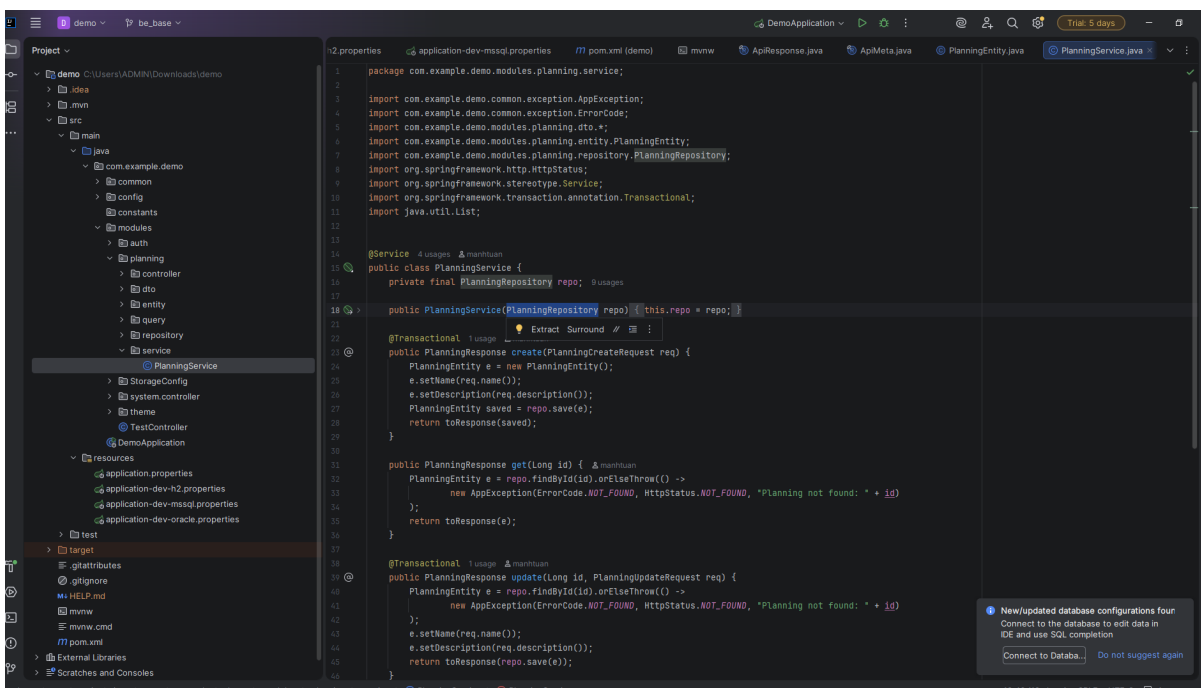
Bước 5: Partner pull code mới về và tiếp tục phát triển

1.5. Ví dụ minh họa

Trường hợp hợp lệ

Partner cần thêm logic trong module planning:

modules/planning/service/PlanningService.java



```
1 package com.example.demo.modules.planning.service;
2
3 import com.example.demo.common.exception.AppException;
4 import com.example.demo.common.exception.ErrorCode;
5 import com.example.demo.modules.planning.dto.*;
6 import com.example.demo.modules.planning.entity.PlanningEntity;
7 import com.example.demo.modules.planning.repository.PlanningRepository;
8 import org.springframework.http.HttpStatus;
9 import org.springframework.stereotype.Service;
10 import org.springframework.transaction.annotation.Transactional;
11 import java.util.List;
12
13
14
15 @Service @usage & marhuan
16 public class PlanningService {
17     private final PlanningRepository repo; @usage
18
19     public PlanningService(PlanningRepository repo) { this.repo = repo; }
20
21
22     @Transactional @usage
23     public PlanningResponse create(PlanningCreateRequest req) {
24         PlanningEntity e = new PlanningEntity();
25         e.setName(req.name());
26         e.setDescription(req.description());
27         PlanningEntity saved = repo.save(e);
28         return toResponse(saved);
29     }
30
31     public PlanningResponse get(Long id) { @marhuan
32         PlanningEntity e = repo.findById(id).orElseThrow() ->
33             new AppException(ErrorCode.NOT_FOUND, HttpStatus.NOT_FOUND, "Planning not found: " + id)
34         );
35         return toResponse(e);
36     }
37
38     @Transactional @usage & marhuan
39     public PlanningResponse update(Long id, PlanningUpdateRequest req) {
40         PlanningEntity e = repo.findById(id).orElseThrow() ->
41             new AppException(ErrorCode.NOT_FOUND, HttpStatus.NOT_FOUND, "Planning not found: " + id)
42         );
43         e.setName(req.name());
44         e.setDescription(req.description());
45         return toResponse(repo.save(e));
46     }
47 }
```

→ Được phép sửa trực tiếp và tạo PR.

Trường hợp không hợp lệ

Partner muốn sửa:

common/security/JwtAuthenticationFilter.java

```
12 import org.springframework.stereotype.Component;
13 import org.springframework.web.filter.OncePerRequestFilter;
14
15 import java.io.IOException;
16 import java.util.List;
17
18 @Component
19 public class JwtAuthenticationFilter extends OncePerRequestFilter {
20
21     private final JwtUtil jwtUtil;
22
23     public JwtAuthenticationFilter(JwtUtil jwtUtil) {
24         this.jwtUtil = jwtUtil;
25     }
26
27     @Override
28     protected void doFilterInternal(HttpServletRequest request,
29                                 HttpServletResponse response,
30                                 FilterChain chain) throws ServletException, IOException {
31
32         String header = request.getHeader("Authorization");
33
34         if (header != null && header.startsWith("Bearer ")) {
35             String token = header.substring(7);
36
37             try {
38                 Claims claims = jwtUtil.parse(token);
39
40                 String username = claims.getSubject();
41                 String role = String.valueOf(claims.get("role"));
42
43                 UsernamePasswordAuthenticationToken auth =
44                     new UsernamePasswordAuthenticationToken(
45                         username,
46                         null,
47                         List.of(new SimpleGrantedAuthority("ROLE_" + role))
48                     );
49
50                 auth.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
51                 SecurityContextHolder.getContext().setAuthentication(auth);
52             } catch (Exception ignored) {
53                 // Token sai/het hạn -> không set auth (API sẽ trả 401)
54             }
55         }
56     }
57 }
```

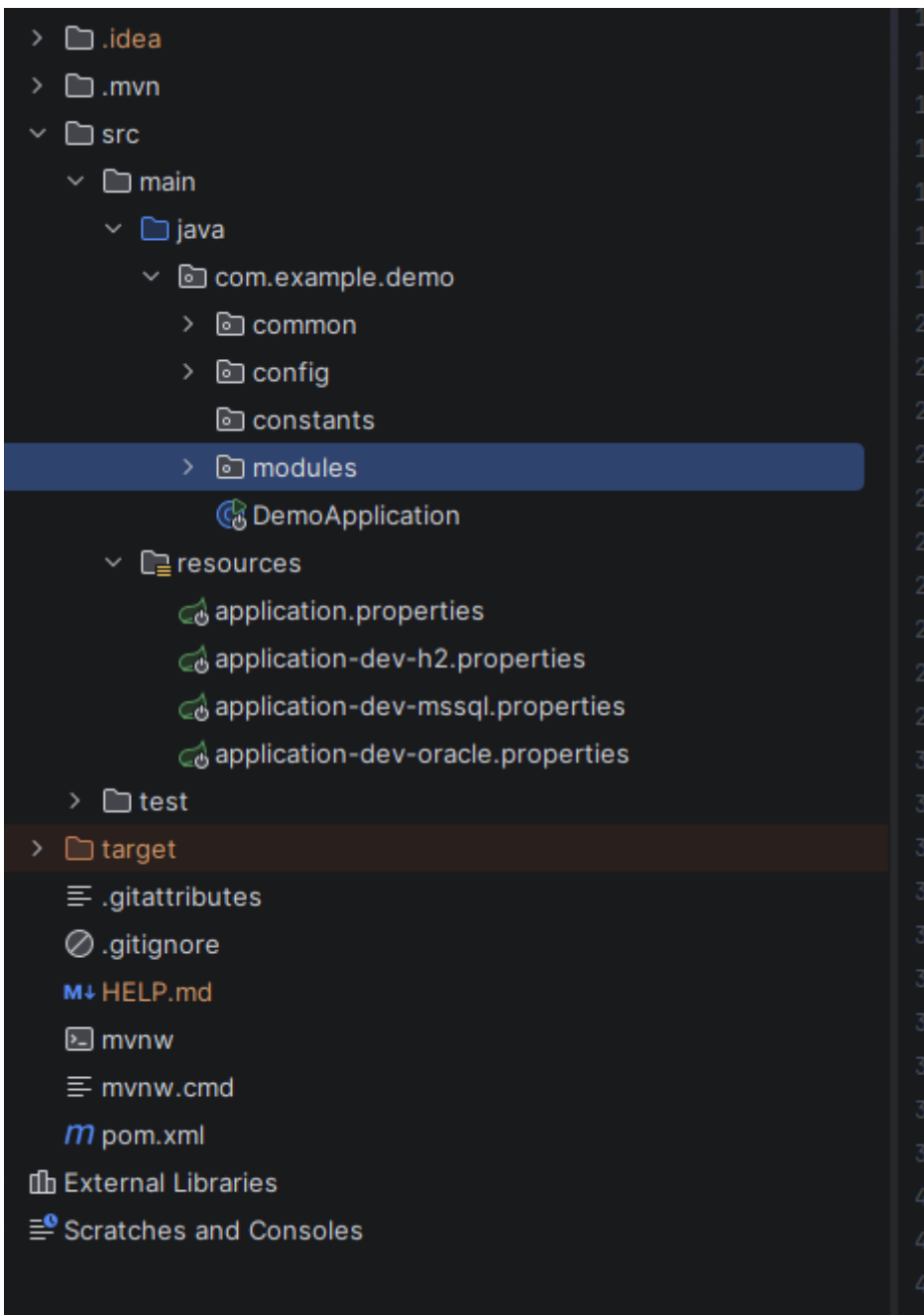
→ Không được sửa trực tiếp.

Phải thực hiện:

1. Tạo Core Change Request
2. Chờ Core Team phê duyệt
3. Core Team thực hiện thay đổi

1.6. Checklist áp dụng

Trước khi commit hoặc tạo PR, cần kiểm tra:



- Code nằm trong modules/**
- Không sửa common/**
- Không sửa config/**
- PR đã được Core Team review (nếu cần)

Phiên bản #1

Được tạo 2026-02-23 10:22:19 UTC bởi admin_lifetex

Được cập nhật 2026-02-23 10:22:19 UTC bởi admin_lifetex