**SAVED JOBS (Lưu tin yêu thích)**

**Entity & Repository:**

"Tạo entity SavedJob với các trường:

- Long id (PK, auto-increment)

- Long userId (FK to users.id, NOT NULL)

- Long jobPostingId (FK to job\_postings.id, NOT NULL)

- LocalDateTime savedAt (audit, NOT NULL, default CURRENT\_TIMESTAMP)

- Unique composite index (user\_id, job\_posting\_id)

- FK constraints với ON DELETE CASCADE

Tạo SavedJobRepository extends JpaRepository<SavedJob, Long>:

- boolean existsByUserIdAndJobPostingId(Long userId, Long jobPostingId)

- Optional<SavedJob> findByUserIdAndJobPostingId(Long userId, Long jobPostingId)

- Page<SavedJob> findByUserIdOrderBySavedAtDesc(Long userId, Pageable pageable)

- void deleteByUserIdAndJobPostingId(Long userId, Long jobPostingId)

- long countByUserId(Long userId)

Viết migration SQL tương ứng."

**Controller:**

"Tạo SavedJobController với 3 endpoints:

1) POST /api/jobs/{jobId}/save

- RBAC: chỉ APPLICANT

- Validate jobId tồn tại và ACTIVE

- Check duplicate bằng existsByUserIdAndJobPostingId, trả 409 nếu đã lưu

- Tạo SavedJob mới, trả 201 với message 'Đã lưu tin tuyển dụng'

- Audit log: user saved job

2) DELETE /api/jobs/{jobId}/unsave

- RBAC: chỉ APPLICANT

- Tìm SavedJob bằng userId + jobPostingId, trả 404 nếu không tìm thấy

- Xóa và trả 200 với message 'Đã bỏ lưu tin tuyển dụng'

- Audit log: user unsaved job

3) GET /api/jobs/saved

- RBAC: chỉ APPLICANT

- Pagination: page=0, size=10, sort=savedAt DESC

- Trả Page<JobPostingResponse> với thông tin đầy đủ job + company

- Thêm trường savedAt trong response

- Chỉ trả jobs còn ACTIVE và chưa hết hạn

Viết unit tests cho cả 3 endpoints: happy path, forbidden, not found, duplicate."

**Cải thiện JobPostingResponse:**

"Trong JobController và PublicJobController, thêm trường isSaved vào JobPostingResponse:

- Nếu user chưa login: isSaved = null

- Nếu user không phải APPLICANT: isSaved = null

- Nếu user là APPLICANT: check existsByUserIdAndJobPostingId và set true/false

Cập nhật method convertToJobPostingResponse để nhận thêm parameter User currentUser và query SavedJobRepository."

**NOTIFICATIONS SYSTEM**

**Entity & Repository:**

"Tạo entity Notification:

- Long id (PK)

- Long userId (FK to users.id, NOT NULL)

- String type (enum: APPLICATION\_STATUS\_CHANGED, NEW\_APPLICATION\_RECEIVED, JOB\_DEADLINE\_REMINDER, INTERVIEW\_SCHEDULED)

- String title (VARCHAR(200), NOT NULL)

- String content (TEXT)

- String actionUrl (VARCHAR(500)) // Link để user click vào

- Boolean isRead (default false, NOT NULL)

- LocalDateTime createdAt (audit, NOT NULL)

- LocalDateTime readAt (nullable)

- Index: (user\_id, is\_read, created\_at DESC)

NotificationRepository:

- Page<Notification> findByUserIdOrderByCreatedAtDesc(Long userId, Pageable pageable)

- Page<Notification> findByUserIdAndIsReadOrderByCreatedAtDesc(Long userId, Boolean isRead, Pageable pageable)

- long countByUserIdAndIsRead(Long userId, Boolean isRead)

- @Modifying @Query UPDATE Notification SET isRead=true, readAt=:readAt WHERE userId=:userId AND isRead=false

- Optional<Notification> findByIdAndUserId(Long id, Long userId)

Viết migration SQL."

**Service:**

"Tạo NotificationService:

1) void createNotification(Long userId, NotificationType type, String title, String content, String actionUrl)

- Tạo Notification mới và save

- Log info: 'Created notification for user {userId}: {title}'

2) void notifyApplicationStatusChanged(Long applicantId, String jobTitle, ApplicationStatus newStatus, String actionUrl)

- title = 'Cập nhật trạng thái đơn ứng tuyển'

- content = 'Trạng thái đơn ứng tuyển cho vị trí {jobTitle} đã được cập nhật thành {status}'

3) void notifyNewApplication(Long employerId, String jobTitle, String applicantName, String actionUrl)

- title = 'Đơn ứng tuyển mới'

- content = '{applicantName} vừa nộp đơn ứng tuyển cho vị trí {jobTitle}'

4) void notifyJobDeadlineReminder(Long applicantId, String jobTitle, int daysLeft, String actionUrl)

- title = 'Tin tuyển dụng sắp hết hạn'

- content = 'Tin tuyển dụng {jobTitle} sẽ hết hạn trong {daysLeft} ngày'

Tích hợp vào ManageApplicationController.updateStatus() và ApplicationController.submit()."

**Controller:**

"Tạo NotificationController:

1) GET /api/notifications

- Pagination: page=0, size=20, sort=createdAt DESC

- Query param: isRead (optional filter)

- Trả Page<NotificationResponse> với đầy đủ thông tin

- RBAC: authenticated user chỉ thấy notification của mình

2) PATCH /api/notifications/{id}/read

- Tìm notification by id và userId, trả 404 nếu không tìm thấy

- Set isRead=true, readAt=now(), save

- Trả 200 với message 'Đã đánh dấu đã đọc'

3) PATCH /api/notifications/mark-all-read

- Update tất cả notification chưa đọc của user thành đã đọc

- Trả 200 với message 'Đã đánh dấu tất cả đã đọc' + count

Viết tests đầy đủ."