

Low-Level Design (LLD)

Low-Level Design Document

Project: {{PROJECT_NAME}} Module/Component: {{MODULE_NAME}}
Version: {{VERSION}} Date: {{DATE}} Author: {{AUTHOR}} Status: Draft |
In Review | Approved Reviewers: {{REVIEWERS}} Related HLD: [HLD Document](#)

Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

1. Module Overview

Module: {{MODULE_NAME}} Service/Repo: {{REPO_NAME}} Team Owner: {{TEAM_NAME}}

Single Responsibility: {{ONE_SENTENCE_WHAT_THIS_MODULE_DOES}}

Boundaries:

- Owns: {{WHAT_THIS_MODULE_OWNS}}
- Does NOT own: {{WHAT_THIS_MODULE_DOES_NOT_OWN}}
- Delegates to: {{WHAT_IT_DELEGATES}}

Key Business Rules:

- {{BUSINESS_RULE_1}}
- {{BUSINESS_RULE_2}}
- {{BUSINESS_RULE_3}}

2. Class / Module Diagram

```
classDiagram
```

```
class {{ServiceClassName}} {  
    -repository: {{RepositoryInterface}}  
    -eventBus: EventBus  
    -logger: Logger  
    +create(dto: Create{{Entity}}Dto): Promise~{{Entity}}~  
    +findById(id: string): Promise~{{Entity}} | null~  
    +findAll(filter: {{Filter}}Dto): Promise~PaginatedResult~{{Entity}}~  
    +update(id: string, dto: Update{{Entity}}Dto): Promise~{{Entity}}~  
    +delete(id: string): Promise~void~  
    -validate(dto: unknown): void  
    -publishEvent(event: {{DomainEvent}}): Promise~void~  
}
```

```
class {{RepositoryInterface}} {  
    <<interface>>  
    +findById(id: string): Promise~{{Entity}} | null~  
    +findMany(filter: {{Filter}}): Promise~{{Entity}}[]~  
    +create(data: Partial~{{Entity}}~): Promise~{{Entity}}~  
    +update(id: string, data: Partial~{{Entity}}~): Promise~{{Entity}}~  
    +delete(id: string): Promise~void~  
    +count(filter: {{Filter}}): Promise~number~  
}
```

```
class {{Entity}} {  
    +id: string  
    +createdAt: Date  
    +updatedAt: Date  
    +deletedAt: Date | null  
    {{FIELD_1}}: {{TYPE_1}}  
    {{FIELD_2}}: {{TYPE_2}}  
    +isDeleted(): boolean  
    +toJSON(): {{EntityJSON}}  
}
```

```
class Create{{Entity}}Dto {  
    {{FIELD_1}}: {{TYPE_1}}
```

```

    {{FIELD_2}}: {{TYPE_2}}
}

class {{Controller}} {
    -service: {{ServiceClassName}}
    +POST /{{resource}}(body: Create{{Entity}}Dto): Response
    +GET /{{resource}}/:id(): Response
    +GET /{{resource}}(query: {{Filter}}Dto): Response
    +PUT /{{resource}}/:id(body: Update{{Entity}}Dto): Response
    +DELETE /{{resource}}/:id(): Response
}

{{ServiceClassName}} --> {{RepositoryInterface}}
{{ServiceClassName}} --> {{Entity}}
{{Controller}} --> {{ServiceClassName}}
{{RepositoryInterface}} ..> {{Entity}}

```

3. Database Schema

3.1 Tables

{{TABLE_NAME_1}}

Purpose: {{TABLE_PURPOSE}}

Column	Type	Nullable	Default	Constraints	Description
id	UUID	NO	gen_random_uuid()	PK	Primary key
created_at	TIMESTAMPTZ	NO	NOW()		Record creation timestamp
updated_at	TIMESTAMPTZ	NO	NOW()		Last update timestamp
deleted_at	TIMESTAMPTZ	YES	NULL		Soft delete timestamp
{{COLUMN_1}}	{{TYPE}}	{{YES/NO}}	{{DEFAULT}}	{{CONSTRAINTS}}	{{DESCRIPTION}}
{{COLUMN_2}}	{{TYPE}}	{{YES/NO}}	{{DEFAULT}}	{{CONSTRAINTS}}	{{DESCRIPTION}}

Column	Type	Nullable	Default	Constraints	Description
{{FK_COLUMN}}	UUID	NO		FK → {{OTHER_TABLE}}(id)	Reference to {{OTHER_TABLE}}}

Indexes:

Index Name	Columns	Type	Rationale
{{TABLE_NAME_1}}_pkey	id	B-tree (PK)	Primary key lookup
idx_{{TABLE_NAME_1}}_{{COLUMN}}	{{COLUMN}}	B-tree	{{QUERY_RATIONALE}}
idx_{{TABLE_NAME_1}}_deleted_at	deleted_at	Partial (WHERE deleted_at IS NULL)	Soft-delete filter performance

{{TABLE_NAME_2}}

Purpose: {{TABLE_PURPOSE}}

Column	Type	Nullable	Default	Constraints	Description
id	UUID	NO	gen_random_uuid()	PK	Primary key
{{COLUMN_1}}	{{TYPE}}	{{YES/NO}}	{{DEFAULT}}	{{CONSTRAINTS}}	{{DESCRIPTION}}

3.2 Enums

```
CREATE TYPE {{ENUM_NAME}} AS ENUM (
  '{{VALUE_1}}',
  '{{VALUE_2}}',
  '{{VALUE_3}}'
);
```

3.3 Migration Notes

- Migration file: {{MIGRATION_FILE_NAME}}.sql
- Zero-downtime: {{YES_NO}} — {{NOTES}}
- Backfill required: {{YES_NO}} — {{BACKFILL_STRATEGY}}
- Estimated migration time: {{ESTIMATE}}

4. API Contract

Base Path: /api/v{{VERSION}}/{{RESOURCE}}

POST /{{resource}}

Summary: Create a new {{entity}}

Authentication: Bearer JWT required

Request Body:

```
{
  "{{field1}}": "{{type_and_example}}",
  "{{field2}}": "{{type_and_example}}"
}
```

Success Response — 201 Created:

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "{{field1}}": "{{value}}",
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

Error Responses:

Status	Code	Description
400	VALIDATION_ERROR	Request body fails validation
401	UNAUTHORIZED	Missing or invalid JWT
403	FORBIDDEN	Insufficient permissions
409	CONFLICT	{{DUPLICATE_FIELD}} already exists
422	BUSINESS_RULE_VIOLATION	{{BUSINESS_RULE}} not met
500	INTERNAL_ERROR	Unexpected server error

GET /{{resource}}/:id

Summary: Retrieve a {{entity}} by ID

Path Parameters:

Parameter	Type	Description
id	UUID	The {{entity}} identifier

Success Response — 200 OK:

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "{{field1}}": "{{value}}",
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

Error Responses:

Status	Code	Description
401	UNAUTHORIZED	Missing or invalid JWT
404	NOT_FOUND	{{entity}} not found

GET /{{resource}}

Summary: List {{entities}} with pagination and filtering

Query Parameters:

Parameter	Type	Default	Description
page	integer	1	Page number (1-based)
limit	integer	20	Items per page (max: 100)
sort	string	createdAt:desc	Sort field and direction
{{filter1}}	{{type}}	—	Filter by {{filter1}}

Success Response — 200 OK:

```
{
  "data": [],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 0,
    "totalPages": 0
  }
}
```

```
}
```

5. Algorithm Specifications

5.1 {{ALGORITHM_NAME}}

Purpose: {{WHAT_IT_DOES}} **Complexity:** Time $O(\{\{TIME_COMPLEXITY\}\})$ | Space $O(\{\{SPACE_COMPLEXITY\}\})$

```
function {{algorithmName}}(input: {{InputType}}): {{OutputType}}
  // Step 1: Validate input
  if input is null or invalid then
    throw ValidationError("{{VALIDATION_MESSAGE}}")
  end if

  // Step 2: {{STEP_2_DESCRIPTION}}
  result = initialize()
  for each item in input.items do
    if item.satisfies({{CONDITION}}) then
      result.add(transform(item))
    end if
  end for

  // Step 3: {{STEP_3_DESCRIPTION}}
  sorted = result.sortBy({{SORT_CRITERIA}})

  // Step 4: Apply business rules
  for each rule in {{BUSINESS_RULES}} do
    sorted = rule.apply(sorted)
  end for

  return sorted
end function
```

Edge Cases:

- **Empty input:** Return empty result, do not throw
- **Single item:** {{SINGLE_ITEM_BEHAVIOR}}
- **Maximum items:** Limit to {{MAX_ITEMS}}, log warning if exceeded

6. Sequence Diagrams

6.1 Create {{Entity}} Flow

```
sequenceDiagram
    autonumber
    actor Client
    participant GW as API Gateway
    participant Auth as Auth Middleware
    participant SVC as {{ServiceClassName}}
    participant DB as Database
    participant MQ as Message Queue
    participant Email as Email Service

    Client->>GW: POST /{{resource}} {body}
    GW->>Auth: Validate JWT
    Auth-->>GW: User context {userId, roles}
    GW->>SVC: create(dto, userContext)
    SVC->>SVC: validate(dto)
    alt Validation fails
        SVC-->>Client: 400 ValidationError
    end
    SVC->>DB: BEGIN TRANSACTION
    SVC->>DB: INSERT INTO {{TABLE_NAME_1}}
    DB-->>SVC: {{entity}} record
    SVC->>DB: COMMIT
    SVC->>MQ: publish("{{ENTITY_CREATED_EVENT}}", event)
    MQ->>Email: Trigger confirmation email
    Email-->>Client: Email delivered async
    SVC-->>GW: {{entity}} dto
    GW-->>Client: 201 Created
```

6.2 {{SECONDARY_FLOW_NAME}} Flow

```
sequenceDiagram
    autonumber
```

```

actor Client
participant SVC as {{ServiceClassName}}
participant Cache as Redis
participant DB as Database

Client->>SVC: GET /{{resource}}/:id
SVC->>Cache: GET {{resource}}:{{id}}
alt Cache hit
    Cache-->>SVC: Cached {{entity}}
    SVC-->>Client: 200 OK (from cache)
else Cache miss
    Cache-->>SVC: null
    SVC->>DB: SELECT * FROM {{TABLE_NAME_1}} WHERE id = $1
    alt Not found
        DB-->>SVC: null
        SVC-->>Client: 404 Not Found
    end
    DB-->>SVC: {{entity}} record
    SVC->>Cache: SET {{resource}}:{{id}} TTL={{CACHE_TTL}}
    SVC-->>Client: 200 OK
end
end

```

7. State Diagrams

```

stateDiagram-v2
    [*] --> {{STATE_DRAFT}} : Created

    {{STATE_DRAFT}} --> {{STATE_PENDING}} : submit()
    {{STATE_DRAFT}} --> [*] : delete()

    {{STATE_PENDING}} --> {{STATE_ACTIVE}} : approve()
    {{STATE_PENDING}} --> {{STATE_REJECTED}} : reject(reason)
    {{STATE_PENDING}} --> {{STATE_DRAFT}} : recall()

    {{STATE_ACTIVE}} --> {{STATE_SUSPENDED}} : suspend(reason)
    {{STATE_ACTIVE}} --> {{STATE_COMPLETED}} : complete()
    {{STATE_ACTIVE}} --> {{STATE_CANCELLED}} : cancel(reason)

```

```

{{STATE_SUSPENDED}} --> {{STATE_ACTIVE}} : reactivate()
{{STATE_SUSPENDED}} --> {{STATE_CANCELLED}} : cancel(reason)

{{STATE_REJECTED}} --> {{STATE_DRAFT}} : revise()
{{STATE_COMPLETED}} --> [*]
{{STATE_CANCELLED}} --> [*]

```

State Transition Rules:

From	To	Trigger	Guard Condition	Side Effect
DRAFT	PENDING	submit()	All required fields populated	Notify reviewers
PENDING	ACTIVE	approve()	Approver has <code>{{PERMISSION}}</code> role	Send welcome email
ACTIVE	SUSPENDED	suspend(reason)	Admin only	Log audit event

8. Error Handling Strategy

8.1 Error Classification

Error Type	HTTP Status	Retry?	Log Level	Alert?
ValidationError	400	No	INFO	No
UnauthorizedError	401	No	WARN	No
ForbiddenError	403	No	WARN	Suspicious patterns only
NotFoundError	404	No	INFO	No
ConflictError	409	No	WARN	No
ExternalServiceError	502	Yes (3x)	ERROR	Yes (if sustained)
DatabaseError	500	Yes (1x)	ERROR	Yes
UnexpectedError	500	No	ERROR	Yes

8.2 Error Response Format (RFC 7807)

```

{
  "type": "https://api.{{DOMAIN}}/errors/{{ERROR_CODE}}",

```

```

"title": "{{Human-readable error title}}",
"status": 400,
"detail": "{{Specific error message}}",
"instance": "/{{resource}}/{{id}}",
"traceId": "{{TRACE_ID}}"
}

```

8.3 Retry & Fallback Strategy

External API call failure:

- Retry with exponential backoff: [1s, 2s, 4s]
- Max retries: 3
- Circuit breaker: Open after 5 failures in 60s window
- Fallback: {{FALLBACK_BEHAVIOR}}
- Alert: PagerDuty if circuit remains open > 5 minutes

9. Concurrency & Thread Safety

Concern	Scenario	Mitigation
Duplicate creation	Two requests create same {{entity}} simultaneously	Unique constraint on {{UNIQUE_FIELD}} + catch constraint error → 409
Optimistic locking	Two updates to same record	version column + check-and-increment
Race condition in {{FLOW}}	{{RACE_CONDITION_DESCRIPTION}}	{{MITIGATION}} (e.g., DB-level lock, Redis SETNX)

10. Performance Considerations

Operation	Target (p99)	Current Baseline	Optimization
GET /{{resource}}/:id	< 50ms	{{BASELINE}}	Redis cache (TTL={{TTL}})
GET /{{resource}} (list)	< 200ms	{{BASELINE}}	Indexed query + cursor pagination
POST /{{resource}}	< 300ms	{{BASELINE}}	Async event publishing
Bulk import (1000 items)	< 5s	{{BASELINE}}	Batch insert in chunks of 100

Known bottlenecks:

- `{{BOTTLENECK_1}}`: `{{MITIGATION}}`
- `{{BOTTLENECK_2}}`: `{{MITIGATION}}`

11. Dependencies

Internal Dependencies

Dependency	Type	Purpose	Fallback if unavailable
<code>{{INTERNAL_SERVICE_1}}</code>	Synchronous	<code>{{PURPOSE}}</code>	<code>{{FALLBACK}}</code>
<code>{{SHARED_LIB}}</code>	Library	<code>{{PURPOSE}}</code>	N/A (required)

External Dependencies

Dependency	Version	Purpose	Fallback if unavailable
PostgreSQL	<code>{{VERSION}}</code>	Primary data store	None — module unavailable
Redis	<code>{{VERSION}}</code>	Caching + session	Degrade gracefully (skip cache)
<code>{{EXTERNAL_API}}</code>	v <code>{{VERSION}}</code>	<code>{{PURPOSE}}</code>	<code>{{FALLBACK}}</code>

12. Configuration Parameters

Variable	Type	Default	Required	Description
<code>{{MODULE_NAME}}_MAX_PAGE_SIZE</code>	integer	100	No	Maximum items per page
<code>{{MODULE_NAME}}_CACHE_TTL_SECONDS</code>	integer	300	No	Cache TTL in seconds
<code>{{MODULE_NAME}}_RETRY_ATTEMPTS</code>	integer	3	No	External API retry count
<code>{{EXTERNAL_API_KEY_VAR}}</code>	string	—	Yes	API key for <code>{{EXTERNAL_SERVICE}}</code>
<code>{{DB_CONNECTION_VAR}}</code>	string	—	Yes	Database connection string

13. Testing Approach

Test Type	Tool	Coverage Target	Location
Unit tests	Jest / Vitest	> {{UNIT_COVERAGE}}%	src/{{module}}/_tests_/unit/
Integration tests	Supertest + TestContainers	Key flows	src/{{module}}/_tests_/integration/
Contract tests	Pact	All public APIs	src/{{module}}/_tests_/contract/

Key test scenarios:

- Create {{entity}} — success path
- Create {{entity}} — validation error (missing required field)
- Create {{entity}} — conflict (duplicate {{UNIQUE_FIELD}})
- Get {{entity}} by ID — found
- Get {{entity}} by ID — not found (404)
- List {{entities}} — with filters, pagination, sorting
- Update {{entity}} — success
- Update {{entity}} — not found
- Delete {{entity}} — success (soft delete)
- State transition — valid transition
- State transition — invalid transition (guard fails)
- External service failure — circuit breaker triggers
- Concurrent creation — duplicate constraint handled

Approval

Role	Name	Date	Signature
Author			
Module Owner			
Security Review			
Tech Lead			

Updated 2026-05-25 07:31:49 UTC by John