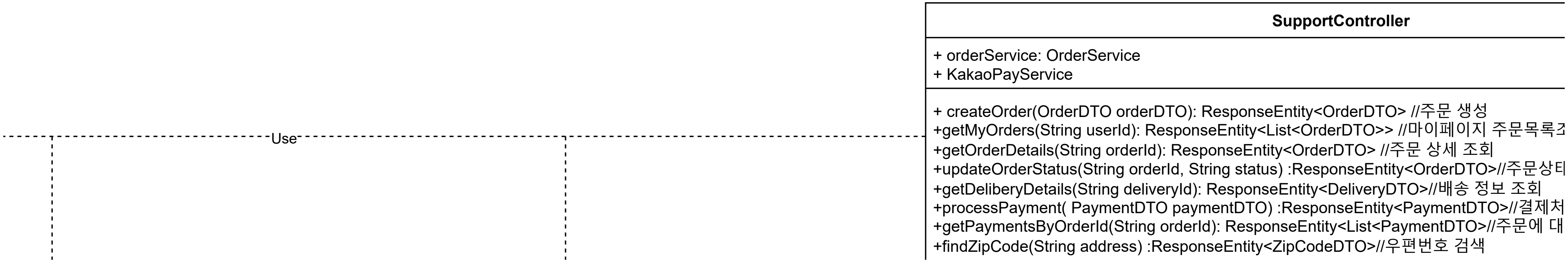
orderandpaym

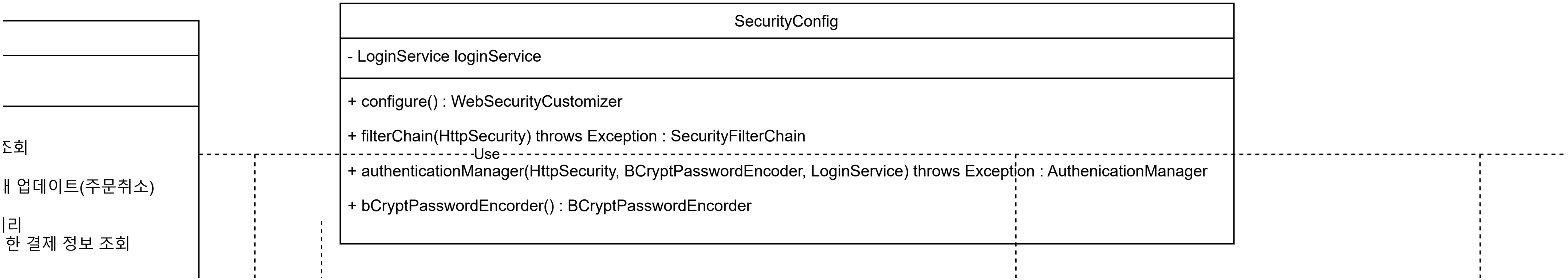


SupportController

+ orderService: OrderService
+ KakaoPayService

+ createOrder(OrderDTO orderDTO): ResponseEntity<OrderDTO> //주문 생성
+getMyOrders(String userId): ResponseEntity<List<OrderDTO>> //마이페이지 주문목록조
+getOrderDetails(String orderId): ResponseEntity<OrderDTO> //주문 상세 조회
+updateOrderStatus(String orderId, String status) :ResponseEntity<OrderDTO>//주문상태
+getDeliberyDetails(String deliveryId): ResponseEntity<DeliveryDTO>//배송 정보 조회
+processPayment(PaymentDTO paymentDTO) :ResponseEntity<PaymentDTO>//결제처
+getPaymentsByOrderId(String orderId): ResponseEntity<List<PaymentDTO>//주문에 대
+findZipCode(String address) :ResponseEntity<ZipCodeDTO>//우편번호 검색

ent (지영영)



SecurityConfig

- LoginService loginService

+ configure() : WebSecurityCustomizer

+ filterChain(HttpSecurity) throws Exception : SecurityFilterChain

+ authenticationManager(HttpSecurity, BCryptPasswordEncoder, LoginService) throws Exception : AuthenicationManager

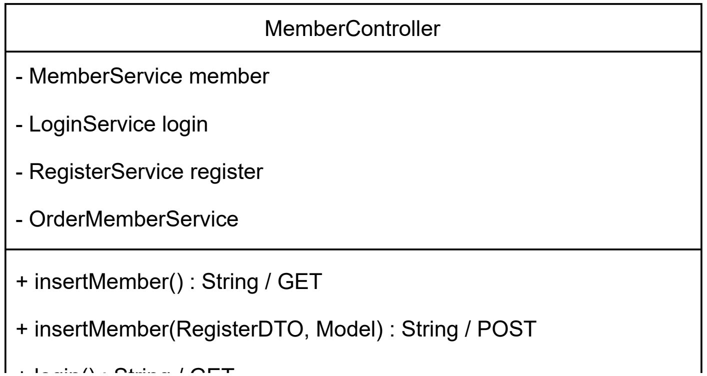
+ bCryptPasswordEncorder() : BCryptPasswordEncorder

4 사용자

. 레파지토리에서 Member로 꺼내서 서비스에
서 DTO에 담는게 통상 쓰는 방식인데 다른 이
유가 있는건지?

1. 레파지토리에서 Member로 꺼내서 서비스에
서 DTO에 담는게 통상 쓰는 방식인데 다른 이
유가 있는건지?

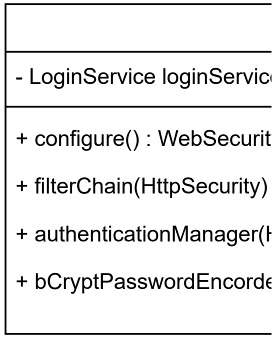
member (세연)



MemberController

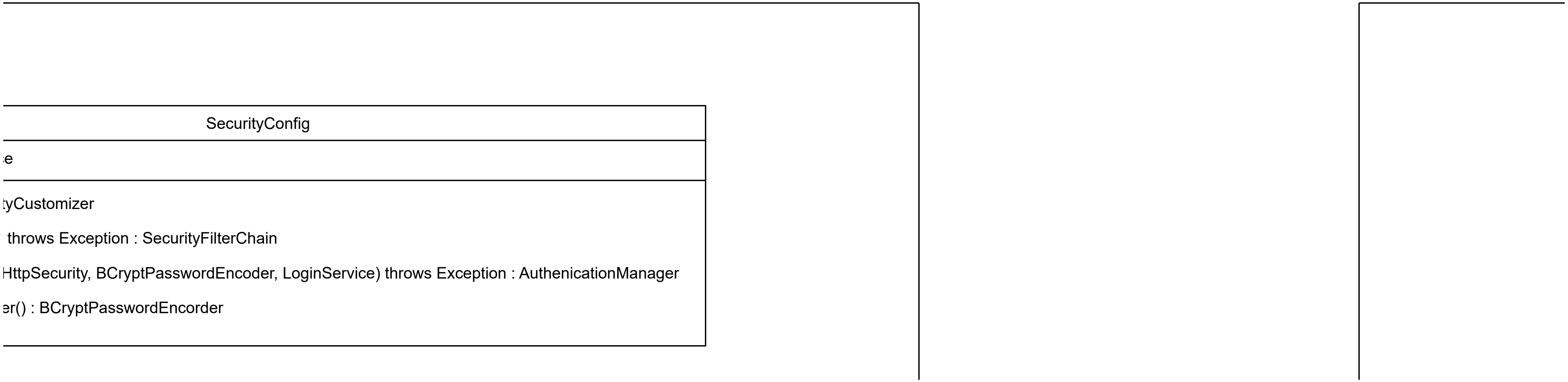
- MemberService member
- LoginService login
- RegisterService register
- OrderMemberService

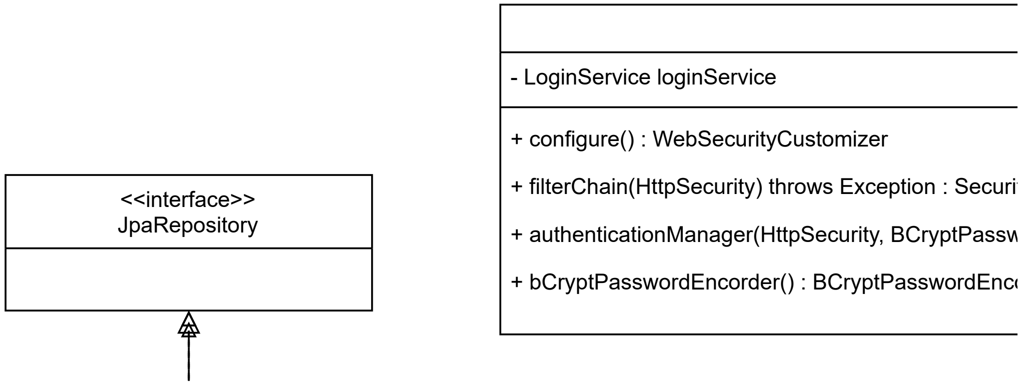
+ insertMember() : String / GET
+ insertMember(RegisterDTO, Model) : String / POST
+ login() : String / GET



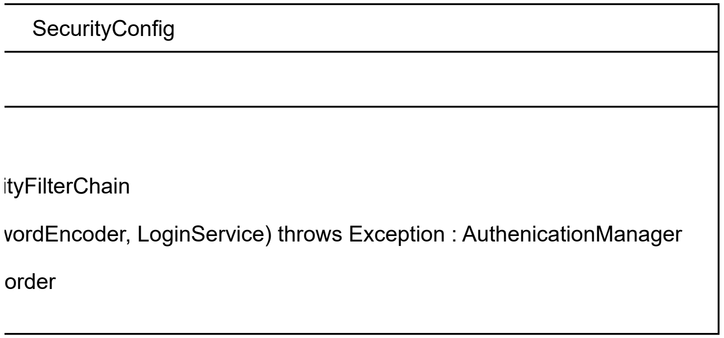
- LoginService loginServic

+ configure() : WebSecurit
+ filterChain(HttpSecurity)
+ authenticationManager(H
+ bCryptPasswordEncode


SecurityConfig
:e
tyCustomizer
throws Exception : SecurityFilterChain
HttpSecurity, BCryptPasswordEncoder, LoginService) throws Exception : AuthenicationManager
er() : BCryptPasswordEncorder


<<interface>>
JpaRepository
- LoginService loginService
+ configure() : WebSecurityCustomizer
+ filterChain(HttpSecurity) throws Exception : Securi
+ authenticationManager(HttpSecurity, BCryptPassw
+ bCryptPasswordEncorder() : BCryptPasswordEnc

project (민희, 주현)

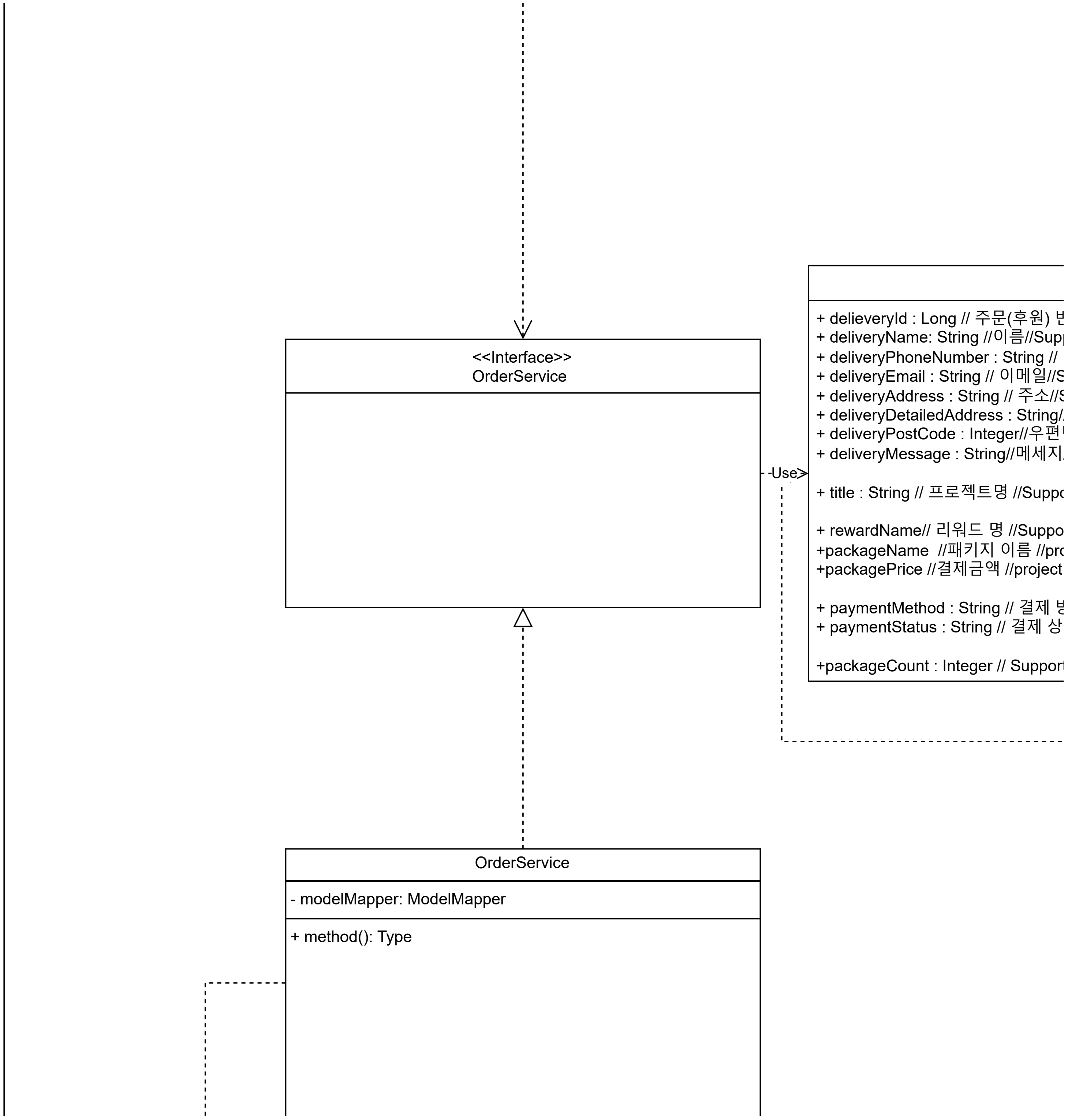


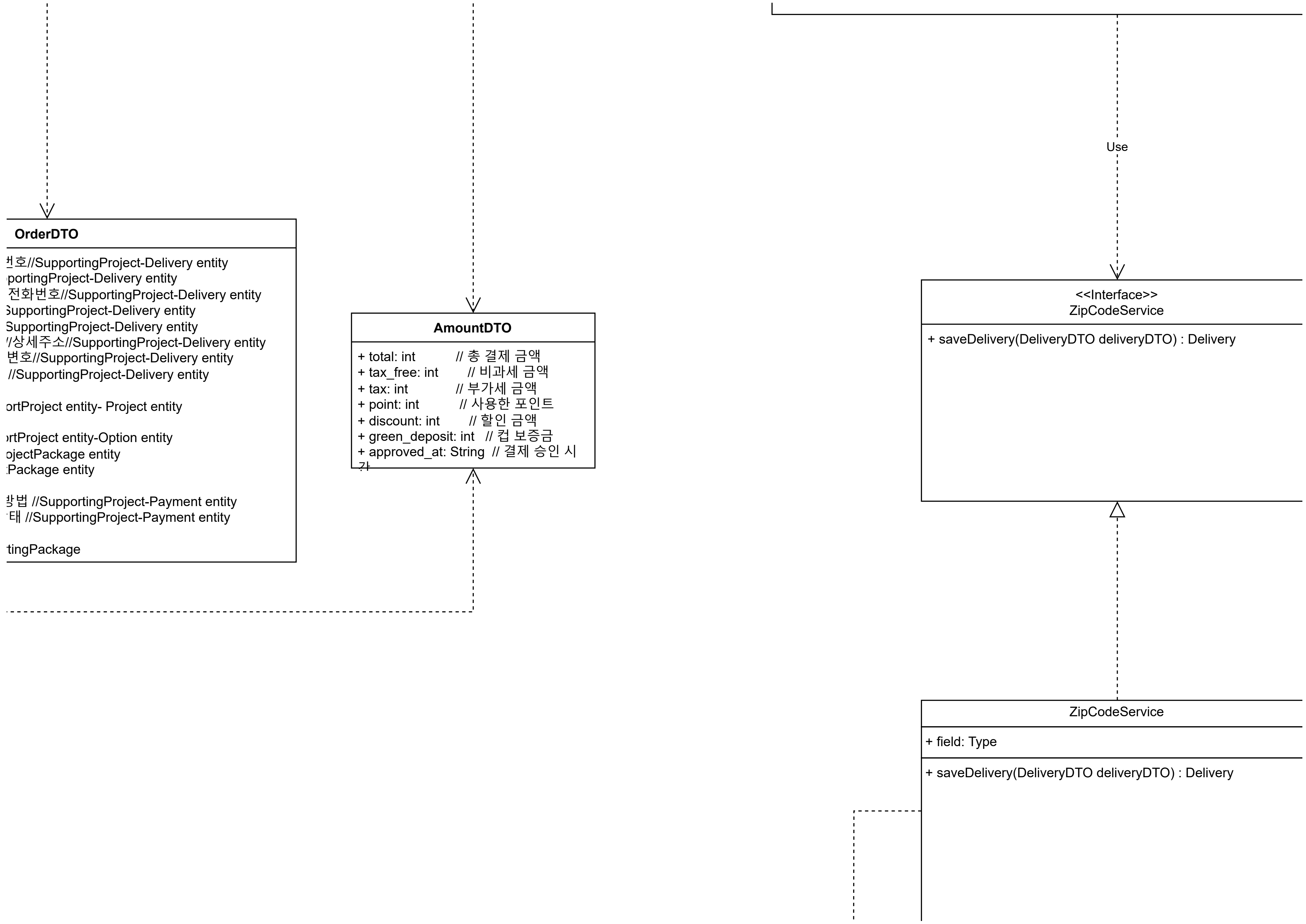
SecurityConfig

ityFilterChain

vordEncoder, LoginService) throws Exception : AuthenicationManager

order


<<Interface>>
OrderService
-Use>
+ delieveryId : Long // 주문(후원) 번
+ deliveryName: String //이름//Sup
+ deliveryPhoneNumber : String //
+ deliveryEmail : String // 이메일//S
+ deliveryAddress : String // 주소//S
+ deliveryDetailedAddress : String/
+ deliveryPostCode : Integer//우편
+ deliveryMessage : String//메세지
+ title : String // 프로젝트명 //Suppo
+ rewardName// 리워드 명 //Suppo
+packageName //패키지 이름 //pro
+packagePrice //결제금액 //project
+ paymentMethod : String // 결제 방
+ paymentStatus : String // 결제 상
+packageCount : Integer // Support
OrderService
- modelMapper: ModelMapper
+ method(): Type



OrderDTO

번호//SupportingProject-Delivery entity
portingProject-Delivery entity
전화번호//SupportingProject-Delivery entity
SupportingProject-Delivery entity
SupportingProject-Delivery entity
/상세주소//SupportingProject-Delivery entity
변호//SupportingProject-Delivery entity
//SupportingProject-Delivery entity

ortProject entity- Project entity

rtProject entity-Option entity
ojectPackage entity
Package entity

방법 //SupportingProject-Payment entity
태 //SupportingProject-Payment entity

tingPackage

AmountDTO

+ total: int // 총 결제 금액
+ tax_free: int // 비과세 금액
+ tax: int // 부가세 금액
+ point: int // 사용한 포인트
+ discount: int // 할인 금액
+ green_deposit: int // 컵 보증금
+ approved_at: String // 결제 승인 시간

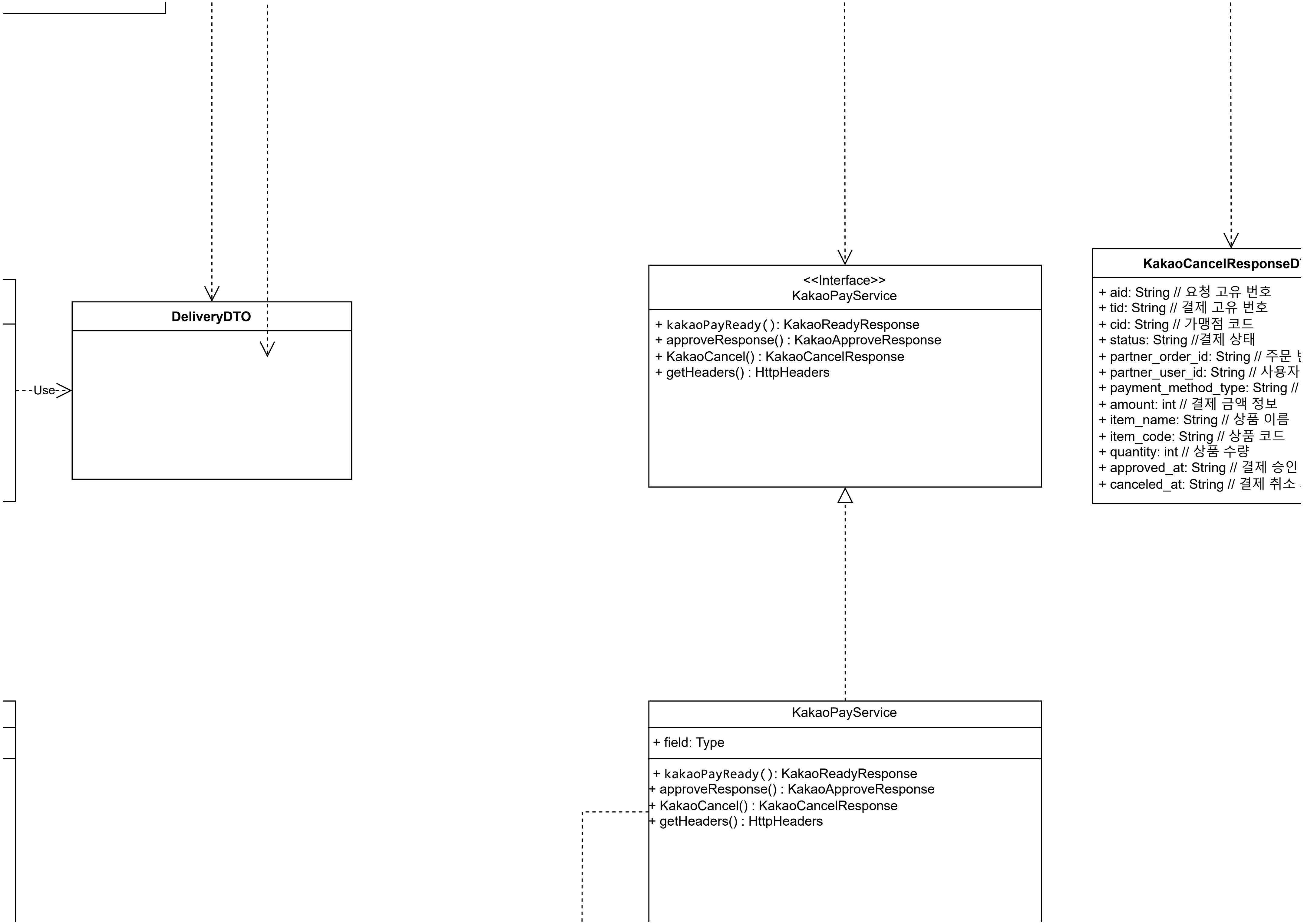
<<Interface>>
ZipCodeService

+ saveDelivery(DeliveryDTO deliveryDTO) : Delivery

ZipCodeService

+ field: Type

+ saveDelivery(DeliveryDTO deliveryDTO) : Delivery



Use

DeliveryDTO

<<Interface>>
KakaoPayService

+ kakaoPayReady(): KakaoReadyResponse
+ approveResponse() : KakaoApproveResponse
+ KakaoCancel() : KakaoCancelResponse
+ getHeaders() : HttpHeaders

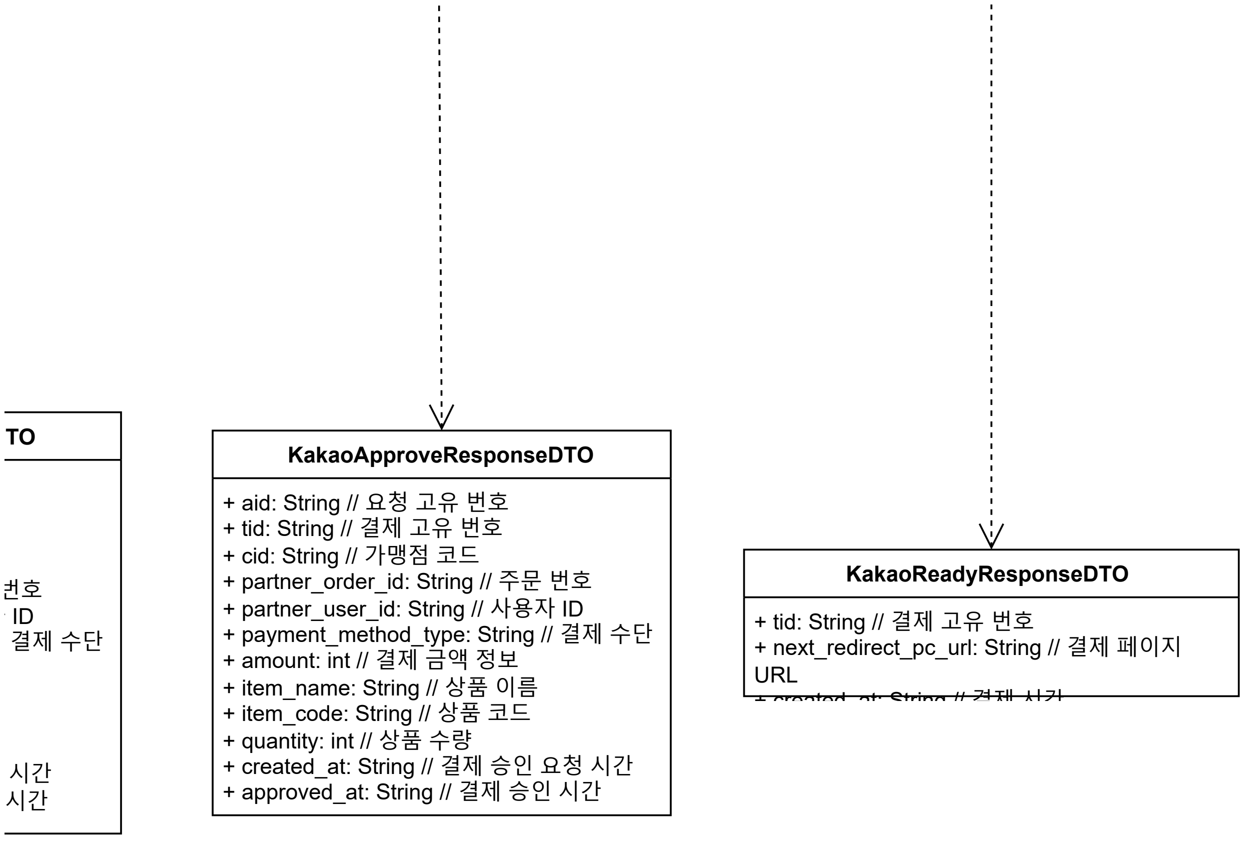
KakaoCancelResponseD

+ aid: String // 요청 고유 번호
+ tid: String // 결제 고유 번호
+ cid: String // 가맹점 코드
+ status: String //결제 상태
+ partner_order_id: String // 주문
+ partner_user_id: String // 사용자
+ payment_method_type: String //
+ amount: int // 결제 금액 정보
+ item_name: String // 상품 이름
+ item_code: String // 상품 코드
+ quantity: int // 상품 수량
+ approved_at: String // 결제 승인
+ canceled_at: String // 결제 취소

KakaoPayService

+ field: Type

+ kakaoPayReady(): KakaoReadyResponse
+ approveResponse() : KakaoApproveResponse
+ KakaoCancel() : KakaoCancelResponse
+ getHeaders() : HttpHeaders



TO

번호
ID
결제 수단

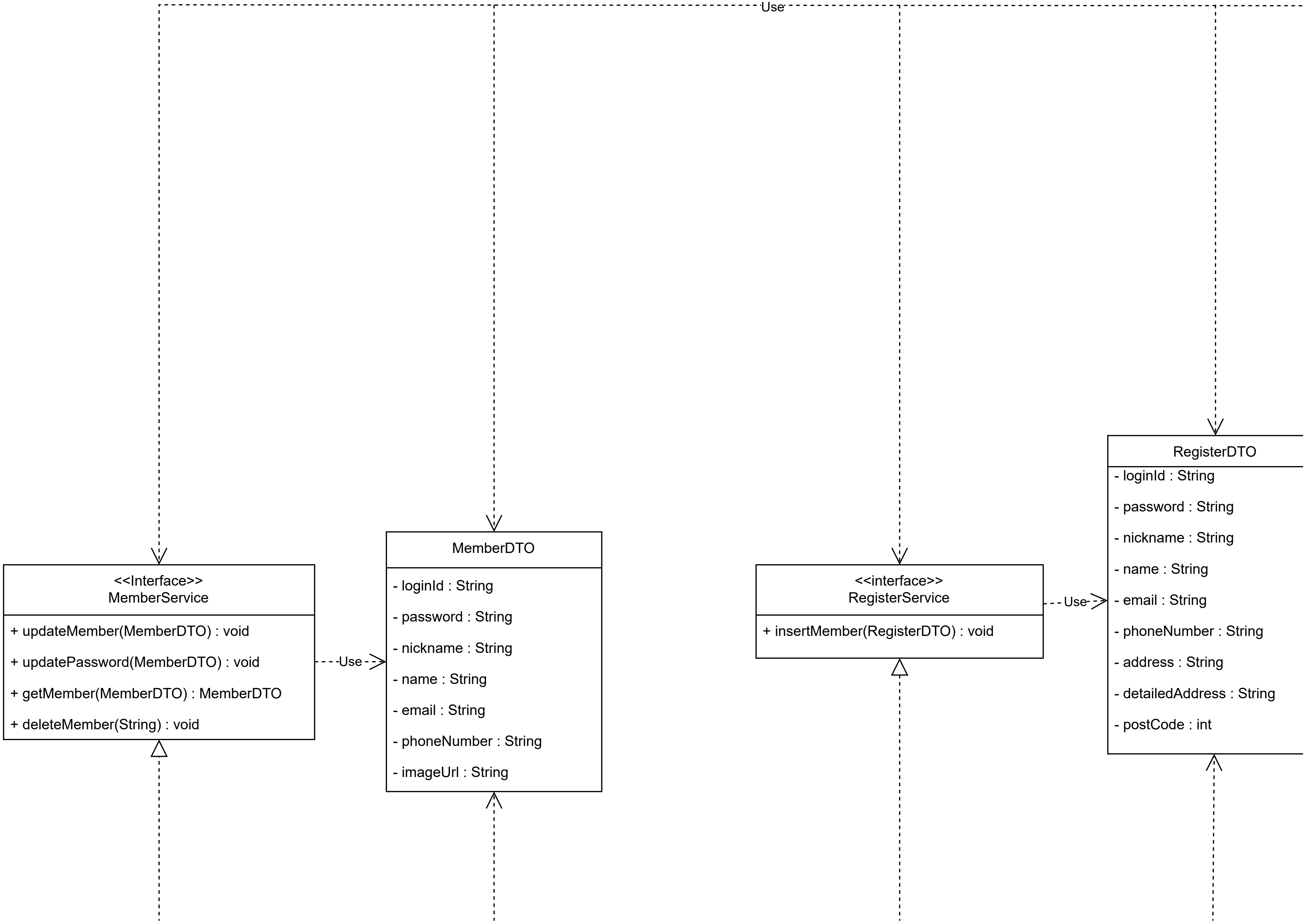
시간
시간

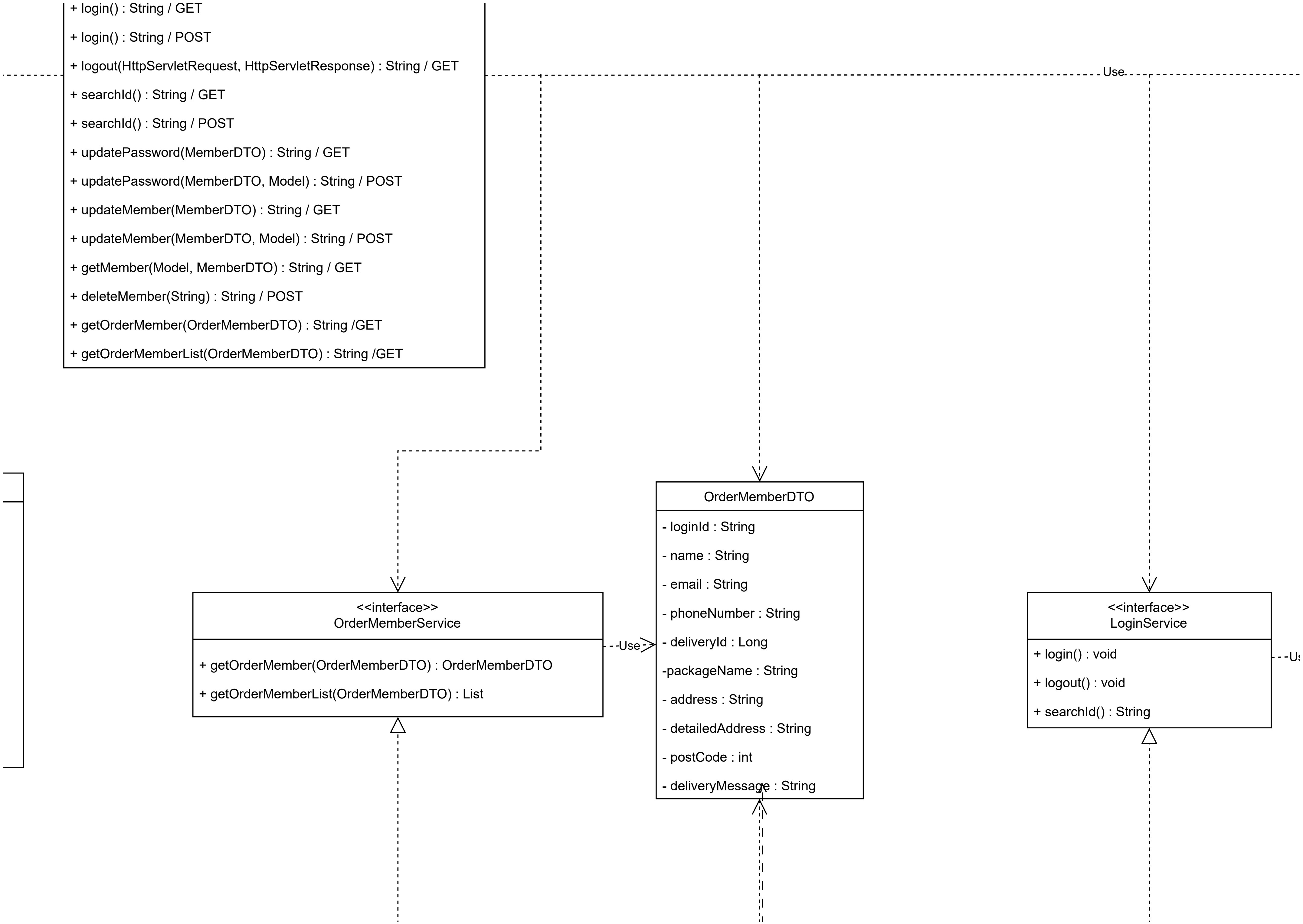
KakaoApproveResponseDTO

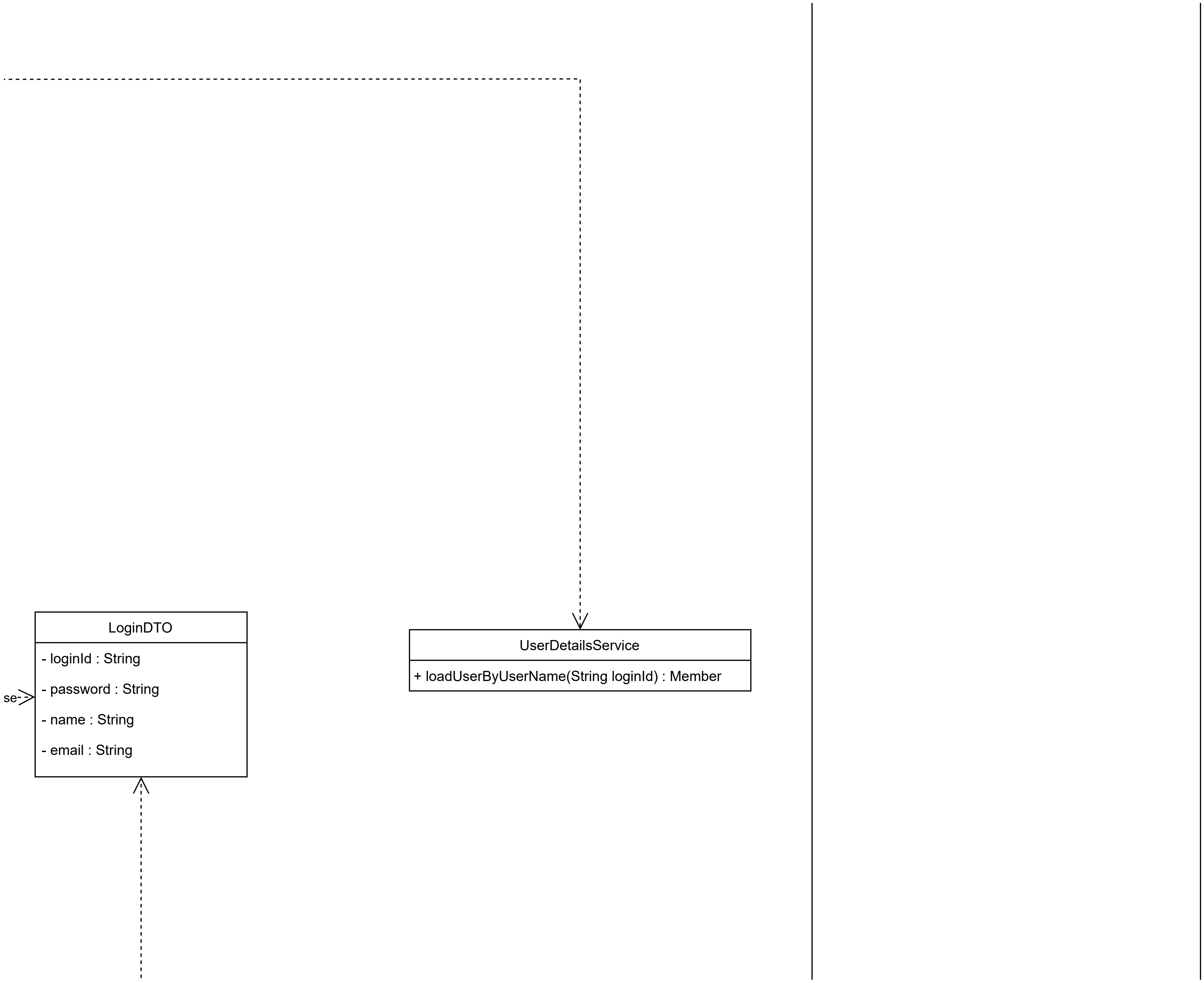
+ aid: String // 요청 고유 번호
+ tid: String // 결제 고유 번호
+ cid: String // 가맹점 코드
+ partner_order_id: String // 주문 번호
+ partner_user_id: String // 사용자 ID
+ payment_method_type: String // 결제 수단
+ amount: int // 결제 금액 정보
+ item_name: String // 상품 이름
+ item_code: String // 상품 코드
+ quantity: int // 상품 수량
+ created_at: String // 결제 승인 요청 시간
+ approved_at: String // 결제 승인 시간

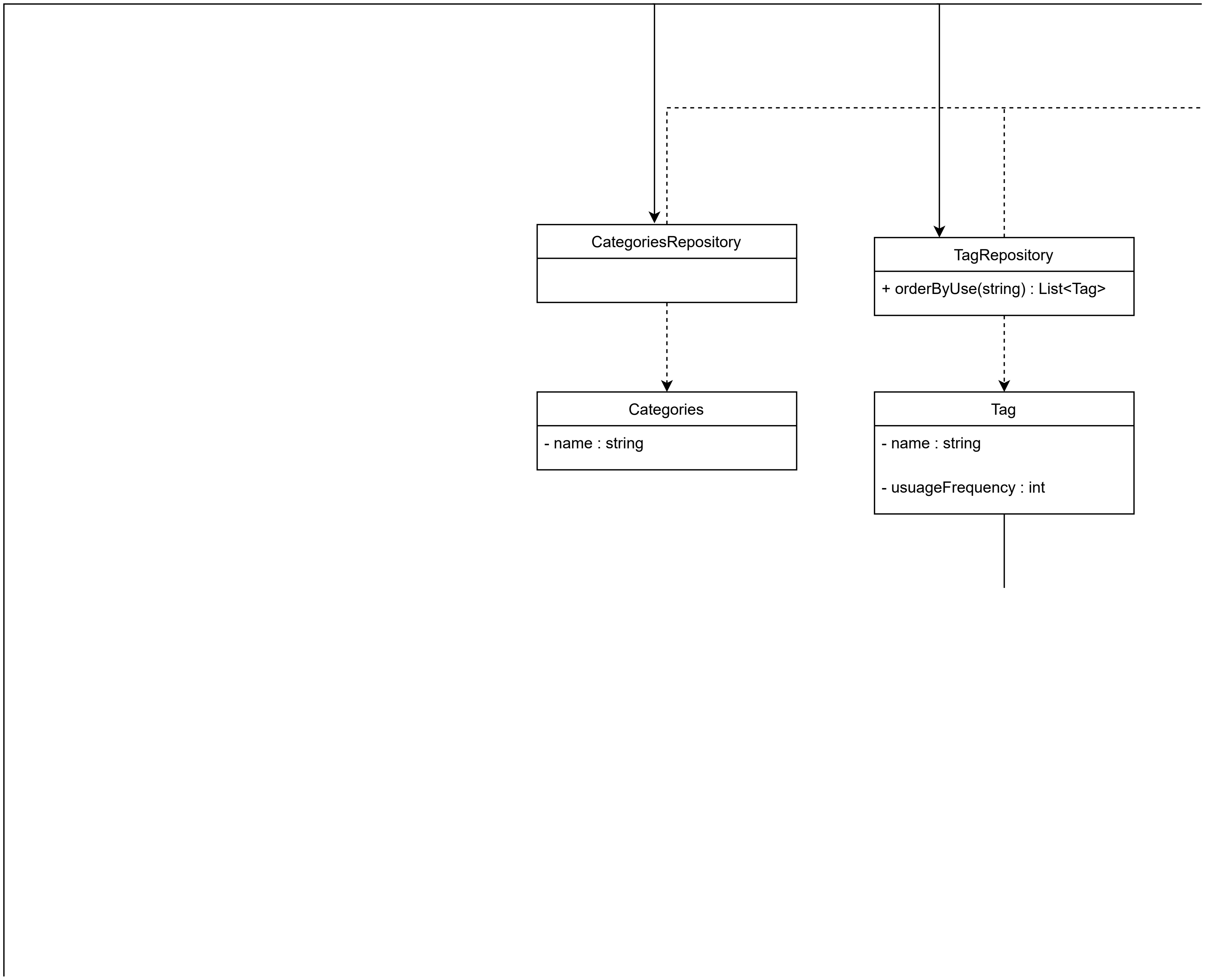
KakaoReadyResponseDTO

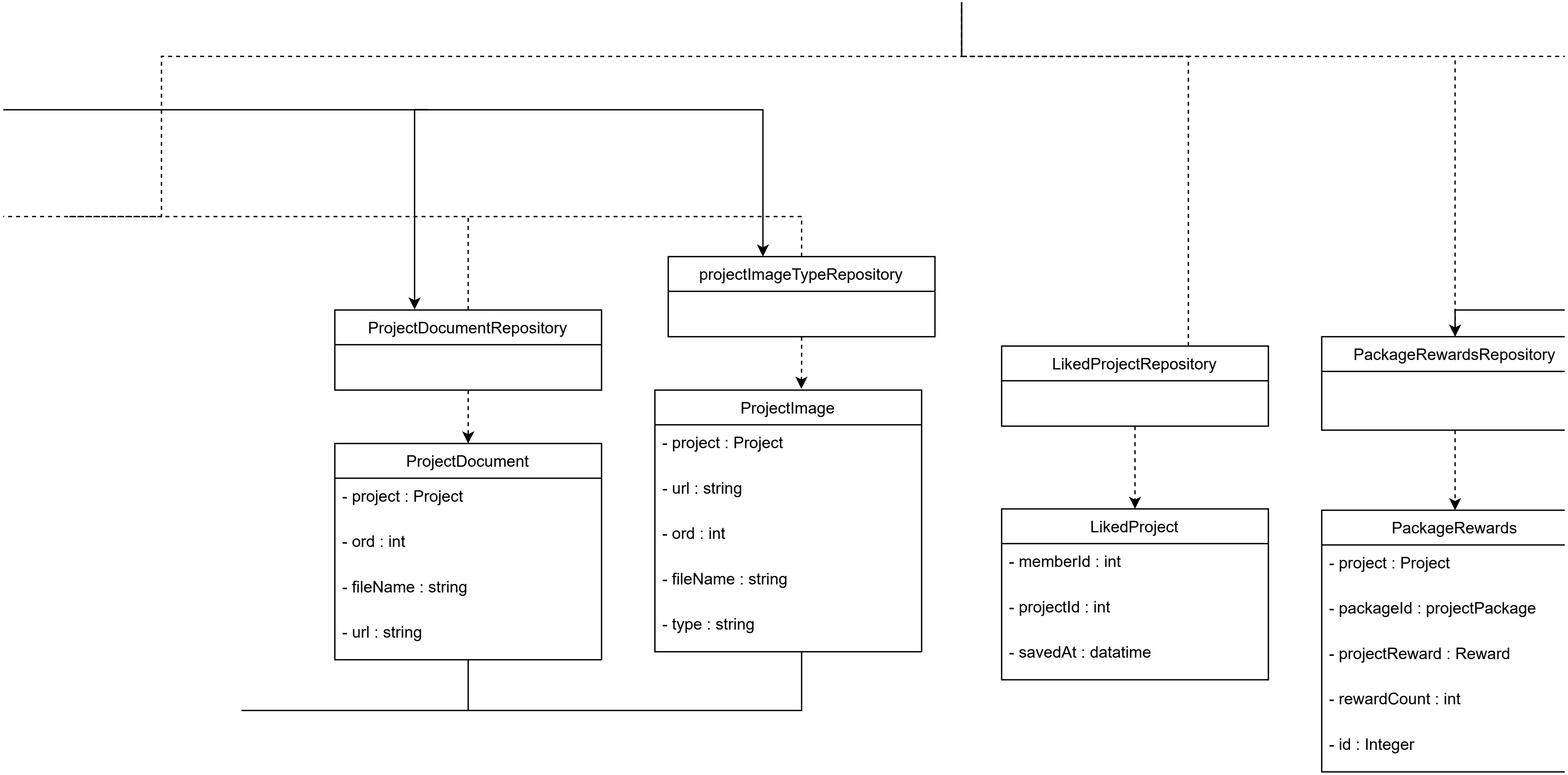
+ tid: String // 결제 고유 번호
+ next_redirect_pc_url: String // 결제 페이지 URL
+ created_at: String // 결제 시간

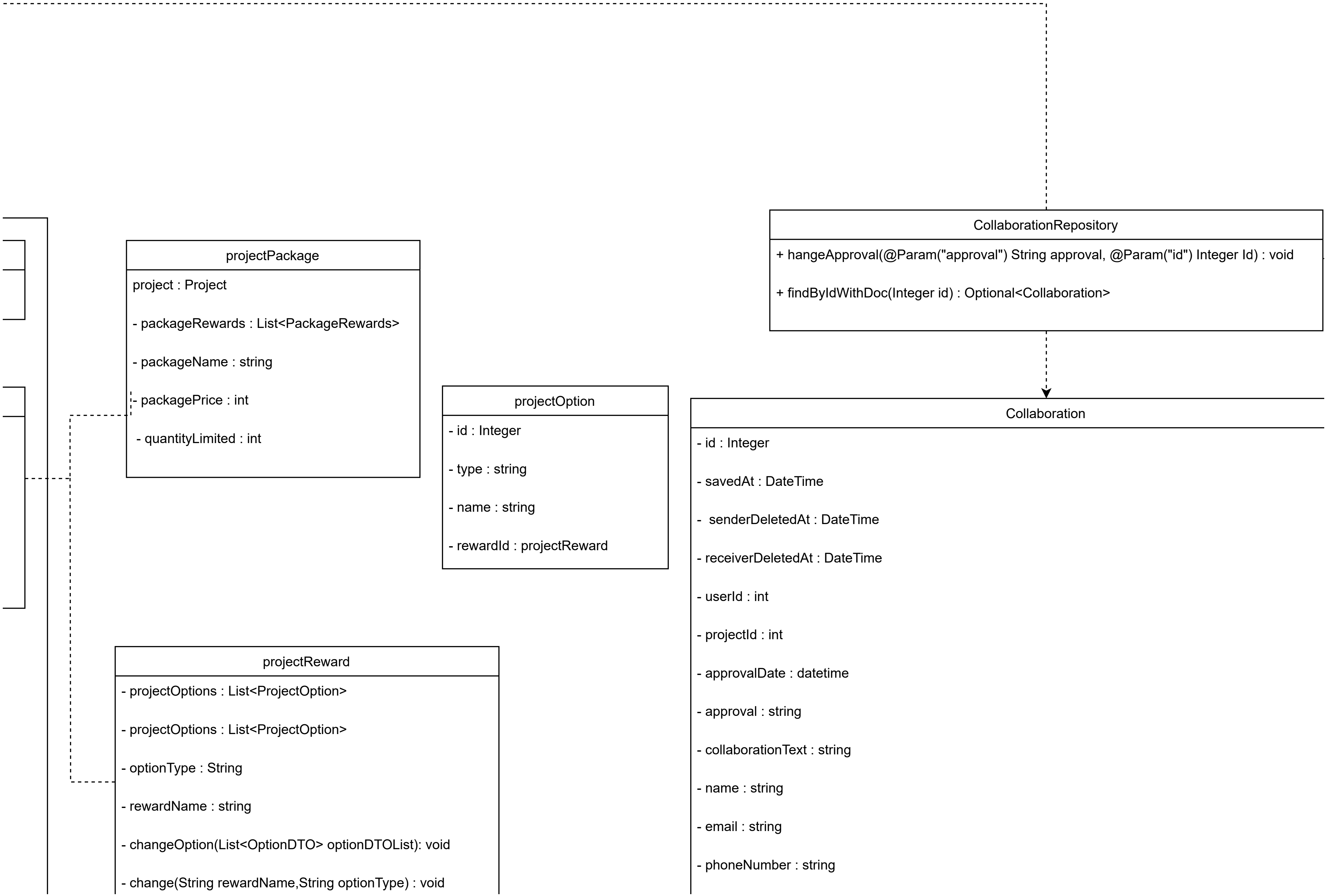

Use
<<Interface>>
MemberService
+ updateMember(MemberDTO) : void
+ updatePassword(MemberDTO) : void
+ getMember(MemberDTO) : MemberDTO
+ deleteMember(String) : void
Use
MemberDTO
- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- imageUrl : String
<<interface>>
RegisterService
+ insertMember(RegisterDTO) : void
Use
RegisterDTO
- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- address : String
- detailedAddress : String
- postCode : int


+ login() : String / GET
+ login() : String / POST
+ logout(HttpServletRequest, HttpServletResponse) : String / GET
+ searchId() : String / GET
+ searchId() : String / POST
+ updatePassword(MemberDTO) : String / GET
+ updatePassword(MemberDTO, Model) : String / POST
+ updateMember(MemberDTO) : String / GET
+ updateMember(MemberDTO, Model) : String / POST
+ getMember(Model, MemberDTO) : String / GET
+ deleteMember(String) : String / POST
+ getOrderMember(OrderMemberDTO) : String /GET
+ getOrderMemberList(OrderMemberDTO) : String /GET
Use
<<interface>>
OrderMemberService
+ getOrderMember(OrderMemberDTO) : OrderMemberDTO
+ getOrderMemberList(OrderMemberDTO) : List
Use
OrderMemberDTO
- loginId : String
- name : String
- email : String
- phoneNumber : String
- deliveryId : Long
-packageName : String
- address : String
- detailedAddress : String
- postCode : int
- deliveryMessage : String
<<interface>>
LoginService
+ login() : void
+ logout() : void
+ searchId() : String
Us


LoginDTO
- loginId : String
- password : String
- name : String
- email : String
se
UserDetailsService
+ loadUserByUserName(String loginId) : Member


CategoriesRepository
TagRepository
+ orderByUse(string) : List<Tag>
Categories
- name : string
Tag
- name : string
- usuageFrequency : int


ProjectDocumentRepository
projectImageTypeRepository
LikedProjectRepository
PackageRewardsRepository
ProjectDocument
- project : Project
- ord : int
- fileName : string
- url : string
ProjectImage
- project : Project
- url : string
- ord : int
- fileName : string
- type : string
LikedProject
- memberId : int
- projectId : int
- savedAt : datatime
PackageRewards
- project : Project
- packageId : projectPackage
- projectReward : Reward
- rewardCount : int
- id : Integer



CollaborationRepository

+ hangeApproval(@Param("approval") String approval, @Param("id") Integer Id) : void

+ findByIdWithDoc(Integer id) : Optional<Collaboration>

projectPackage

project : Project

- packageRewards : List<PackageRewards>

- packageName : string

- packagePrice : int

- quantityLimited : int

projectOption

- id : Integer

- type : string

- name : string

- rewardId : projectReward

Collaboration

- id : Integer

- savedAt : DateTime

- senderDeletedAt : DateTime

- receiverDeletedAt : DateTime

- userId : int

- projectId : int

- approvalDate : datetime

- approval : string

- collaborationText : string

- name : string

- email : string

- phoneNumber : string

projectReward

- projectOptions : List<ProjectOption>

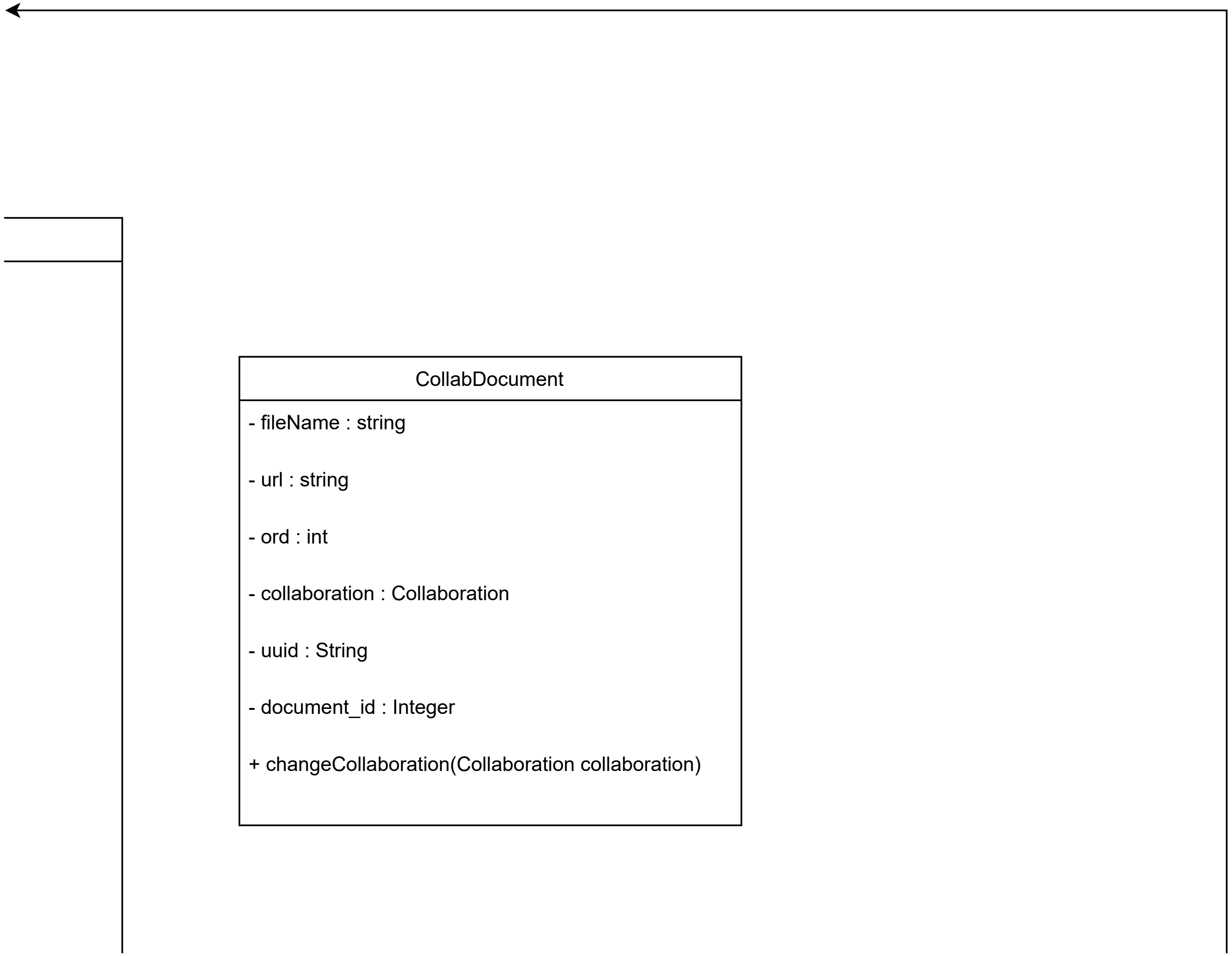
- projectOptions : List<ProjectOption>

- optionType : String

- rewardName : string

- changeOption(List<OptionDTO> optionDTOList): void

- change(String rewardName,String optionType) : void



CollabDocument

- fileName : string

- url : string

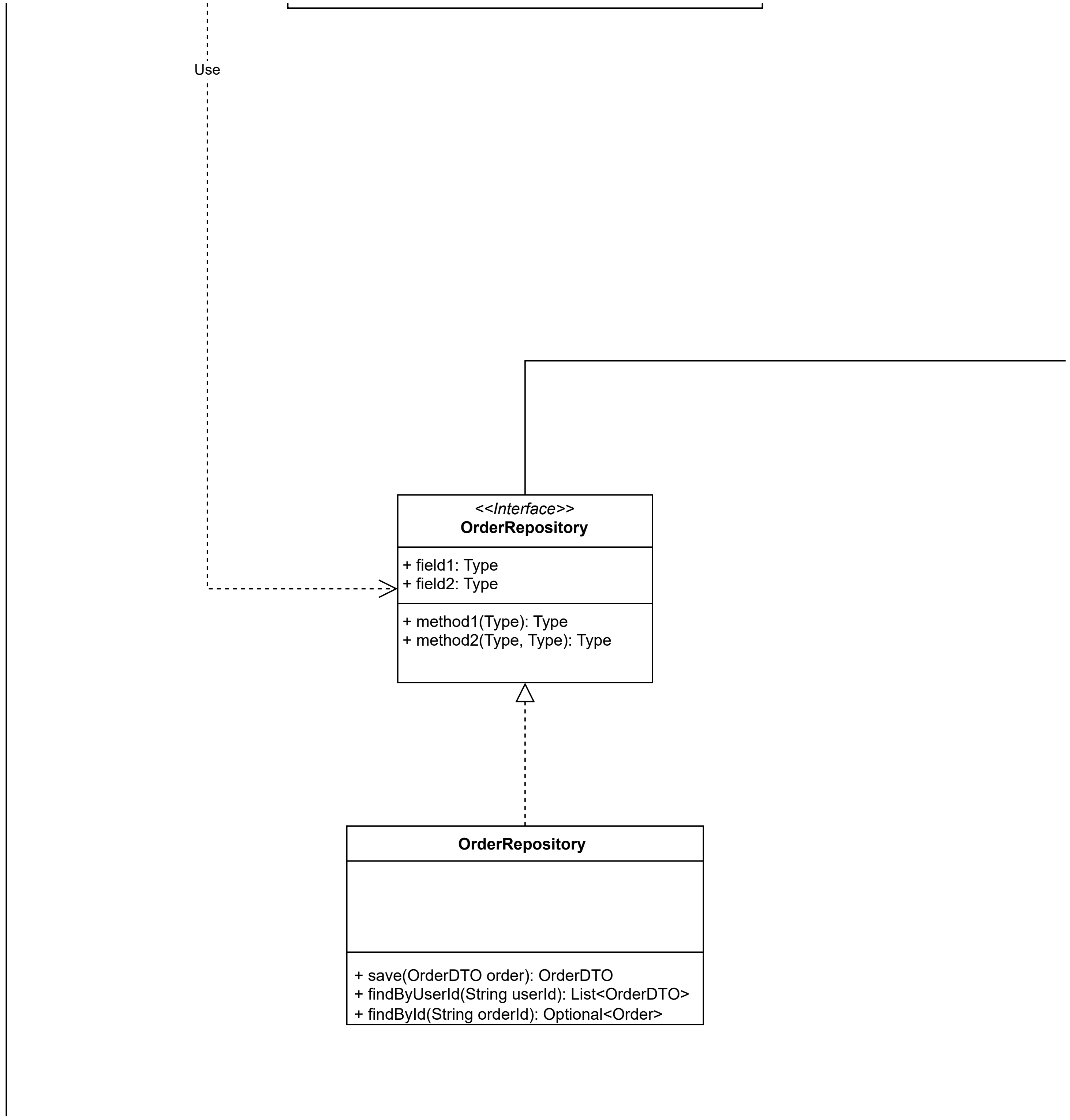
- ord : int

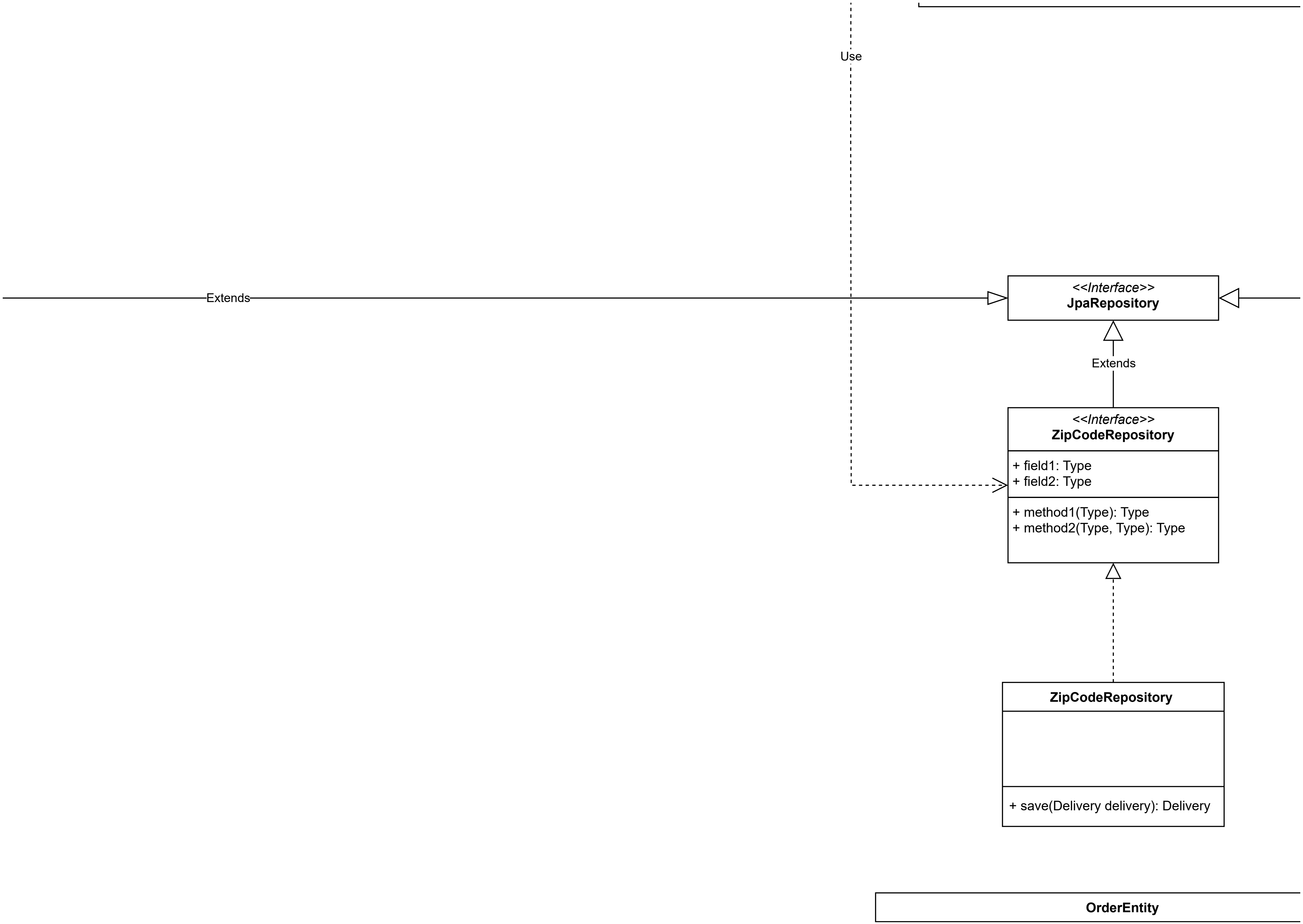
- collaboration : Collaboration

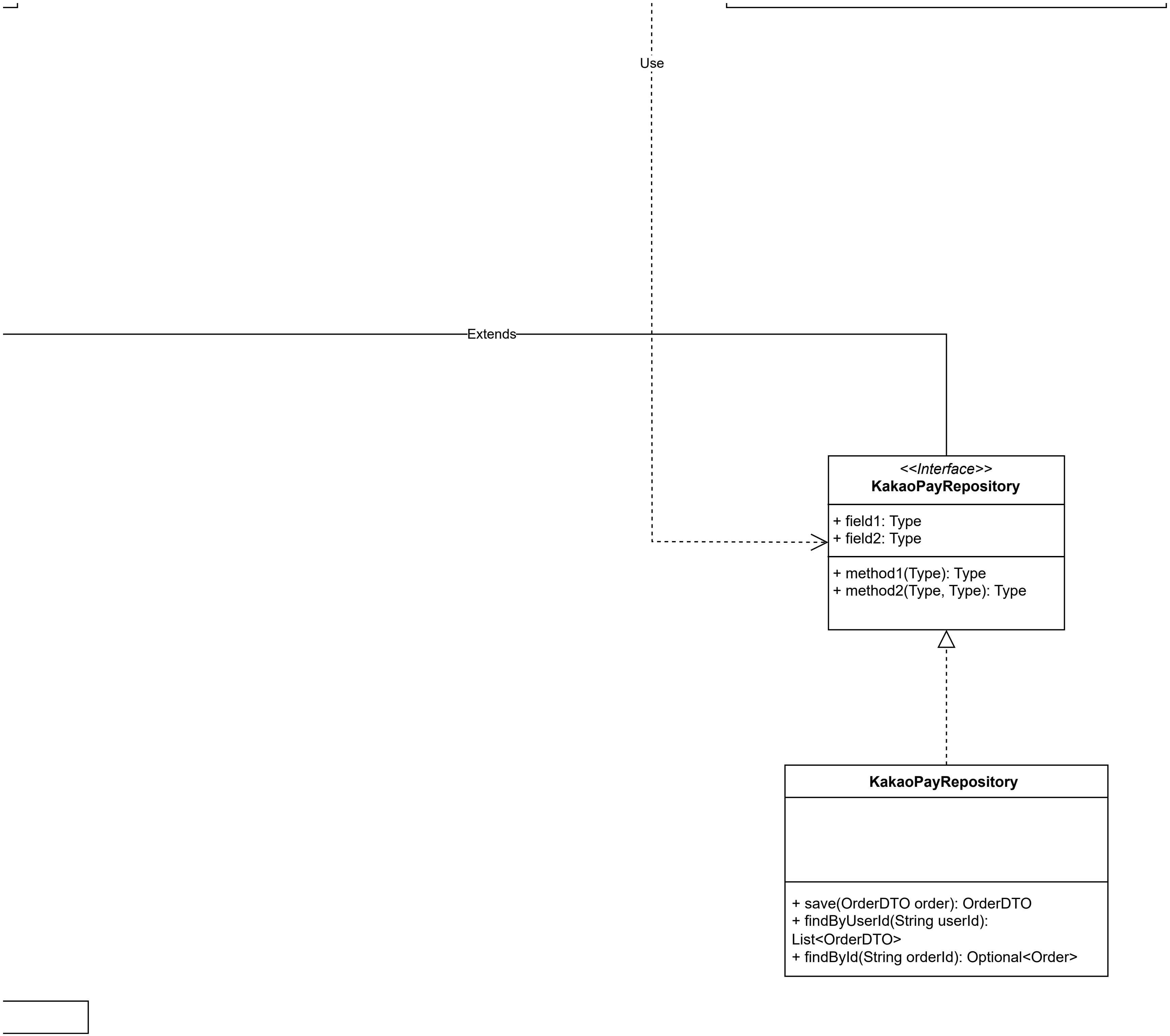
- uuid : String

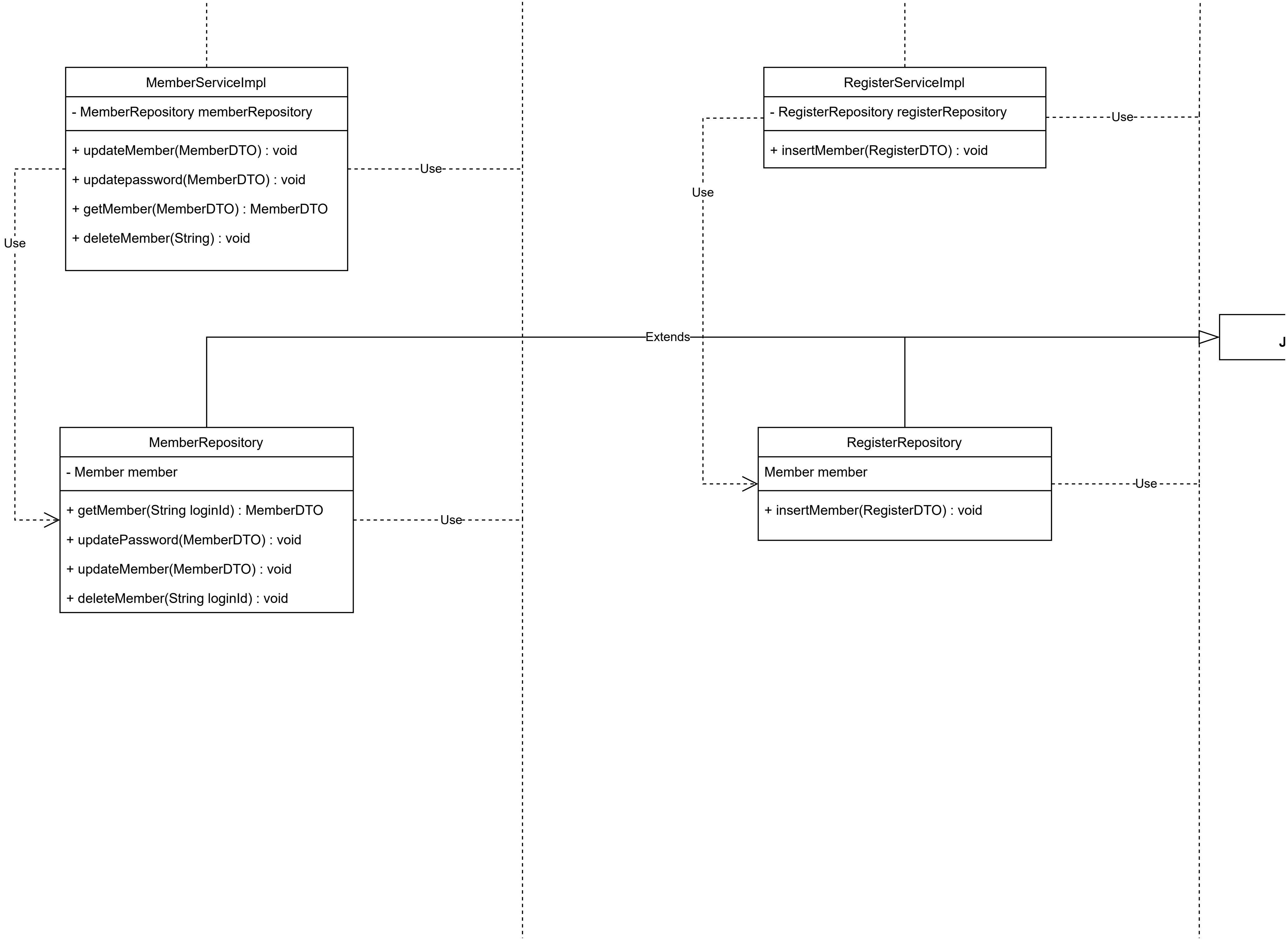
- document_id : Integer

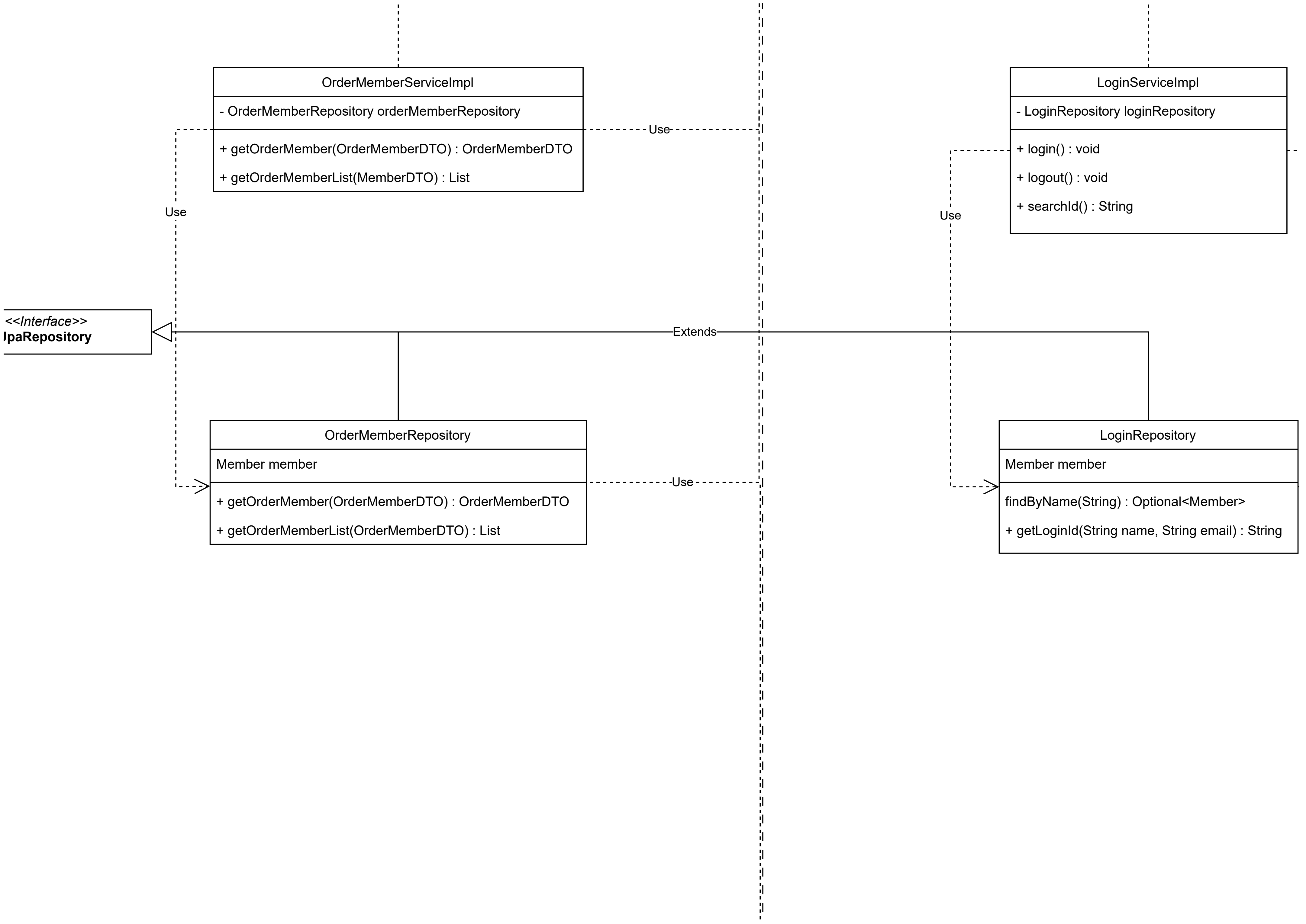
+ changeCollaboration(Collaboration collaboration)


Use
<<Interface>>
OrderRepository
+ field1: Type
+ field2: Type
+ method1(Type): Type
+ method2(Type, Type): Type
OrderRepository
+ save(OrderDTO order): OrderDTO
+ findByUserId(String userId): List<OrderDTO>
+ findById(String orderId): Optional<Order>


Use
Extends
<<Interface>>
JpaRepository
Extends
<<Interface>>
ZipCodeRepository
+ field1: Type
+ field2: Type
+ method1(Type): Type
+ method2(Type, Type): Type
ZipCodeRepository
+ save(Delivery delivery): Delivery
OrderEntity

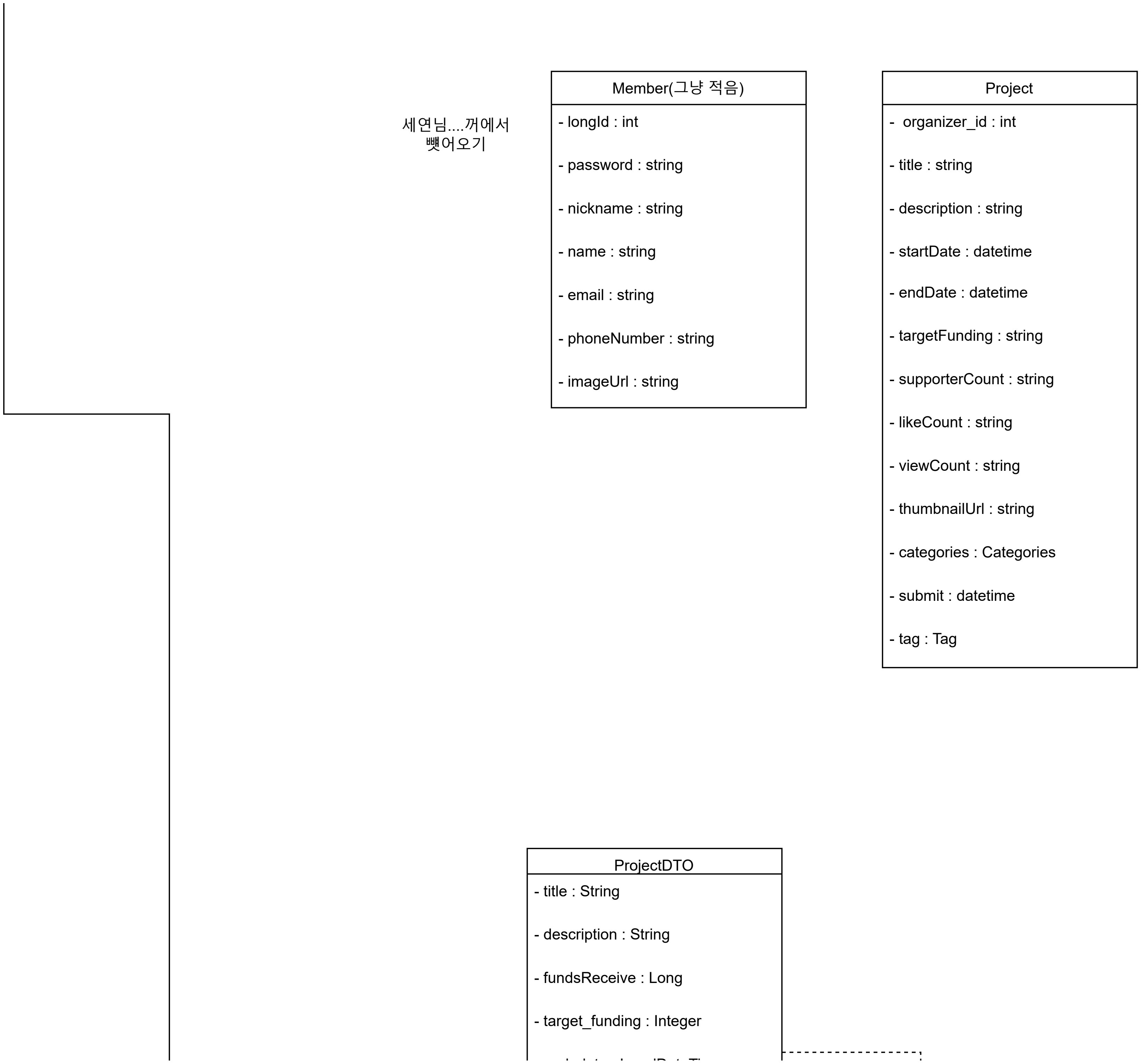

Use
Extends
<<Interface>>
KakaoPayRepository
+ field1: Type
+ field2: Type
+ method1(Type): Type
+ method2(Type, Type): Type
KakaoPayRepository
+ save(OrderDTO order): OrderDTO
+ findByUserId(String userId): List<OrderDTO>
+ findById(String orderId): Optional<Order>


MemberServiceImpl
- MemberRepository memberRepository
+ updateMember(MemberDTO) : void
+ updatepassword(MemberDTO) : void
+ getMember(MemberDTO) : MemberDTO
+ deleteMember(String) : void
Use
Use
RegisterServiceImpl
- RegisterRepository registerRepository
+ insertMember(RegisterDTO) : void
Use
Use
Extends
J
MemberRepository
- Member member
+ getMember(String loginId) : MemberDTO
+ updatePassword(MemberDTO) : void
+ updateMember(MemberDTO) : void
+ deleteMember(String loginId) : void
Use
RegisterRepository
Member member
+ insertMember(RegisterDTO) : void
Use


OrderMemberServiceImpl
- OrderMemberRepository orderMemberRepository
+ getOrderMember(OrderMemberDTO) : OrderMemberDTO
+ getOrderMemberList(MemberDTO) : List
Use
Use
LoginServiceImpl
- LoginRepository loginRepository
+ login() : void
+ logout() : void
+ searchId() : String
Use
<<Interface>>
JpaRepository
Extends
OrderMemberRepository
Member member
+ getOrderMember(OrderMemberDTO) : OrderMemberDTO
+ getOrderMemberList(OrderMemberDTO) : List
Use
LoginRepository
Member member
findByName(String) : Optional<Member>
+ getLoginId(String name, String email) : String

Use

Use



세연님....꺼에서
뺏어오기

Member(그냥 적음)

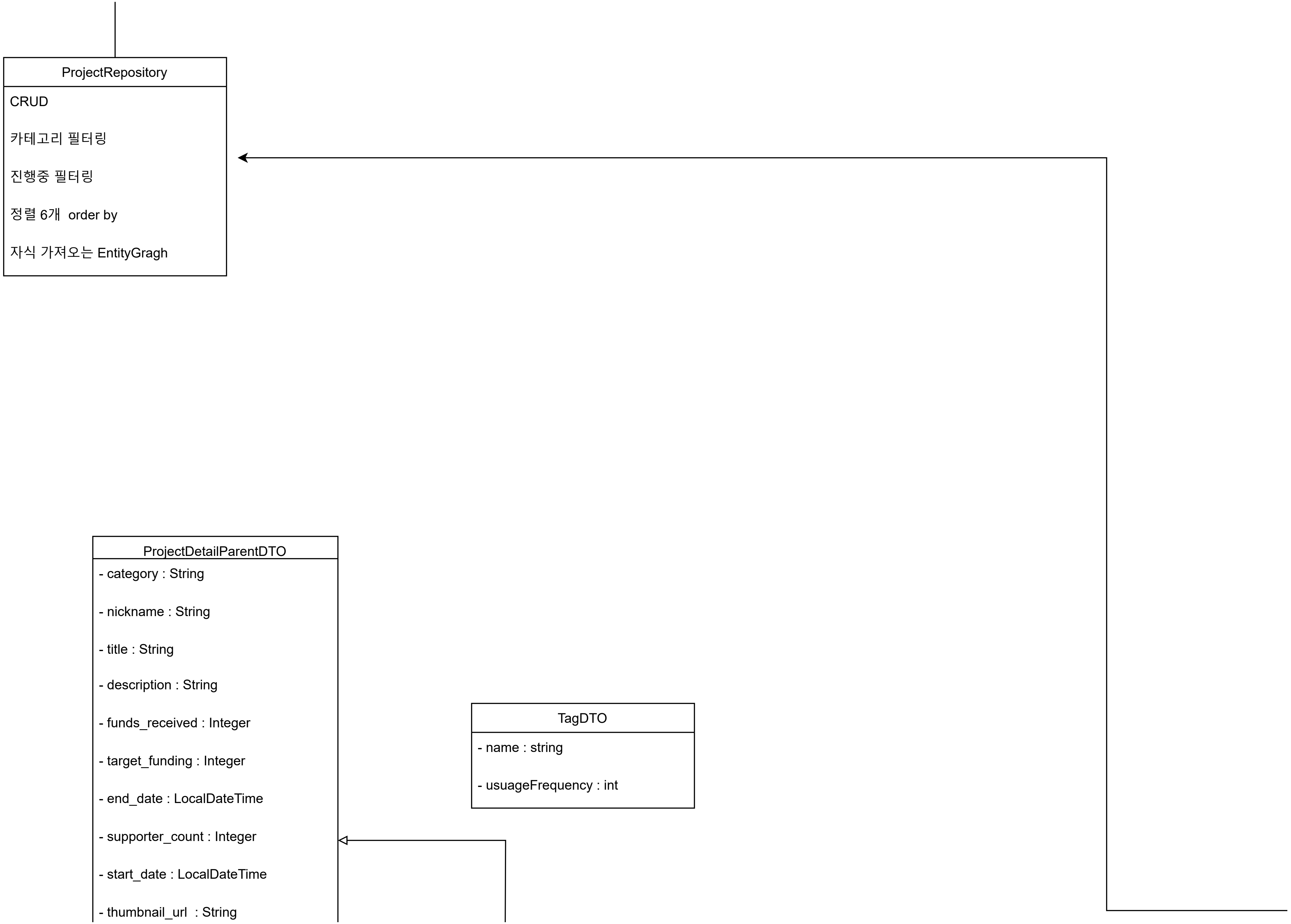
- longId : int
- password : string
- nickname : string
- name : string
- email : string
- phoneNumber : string
- imageUrl : string

Project

- organizer_id : int
- title : string
- description : string
- startDate : datetime
- endDate : datetime
- targetFunding : string
- supporterCount : string
- likeCount : string
- viewCount : string
- thumbnailUrl : string
- categories : Categories
- submit : datetime
- tag : Tag

ProjectDTO

- title : String
- description : String
- fundsReceive : Long
- target_funding : Integer



ProjectRepository

CRUD

카테고리 필터링

진행중 필터링

정렬 6개 order by

자식 가져오는 EntityGragh

ProjectDetailParentDTO

- category : String

- nickname : String

- title : String

- description : String

- funds_received : Integer

- target_funding : Integer

- end_date : LocalDateTime

- supporter_count : Integer

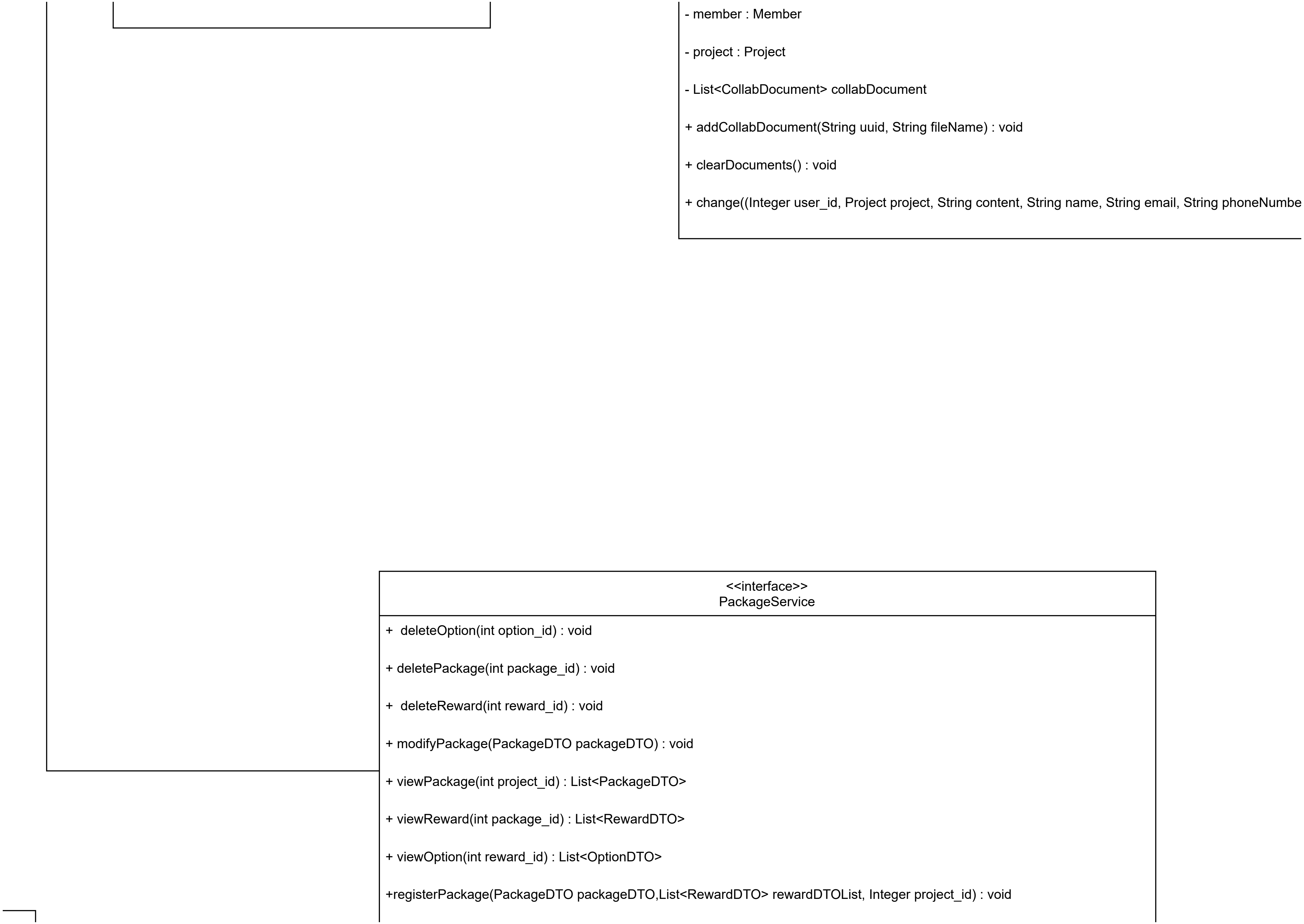
- start_date : LocalDateTime

- thumbnail_url : String

TagDTO

- name : string

- usuageFrequency : int



- member : Member

- project : Project

- List<CollabDocument> collabDocument

+ addCollabDocument(String uuid, String fileName) : void

+ clearDocuments() : void

+ change((Integer user_id, Project project, String content, String name, String email, String phoneNumbe

<<interface>>
PackageService

+ deleteOption(int option_id) : void

+ deletePackage(int package_id) : void

+ deleteReward(int reward_id) : void

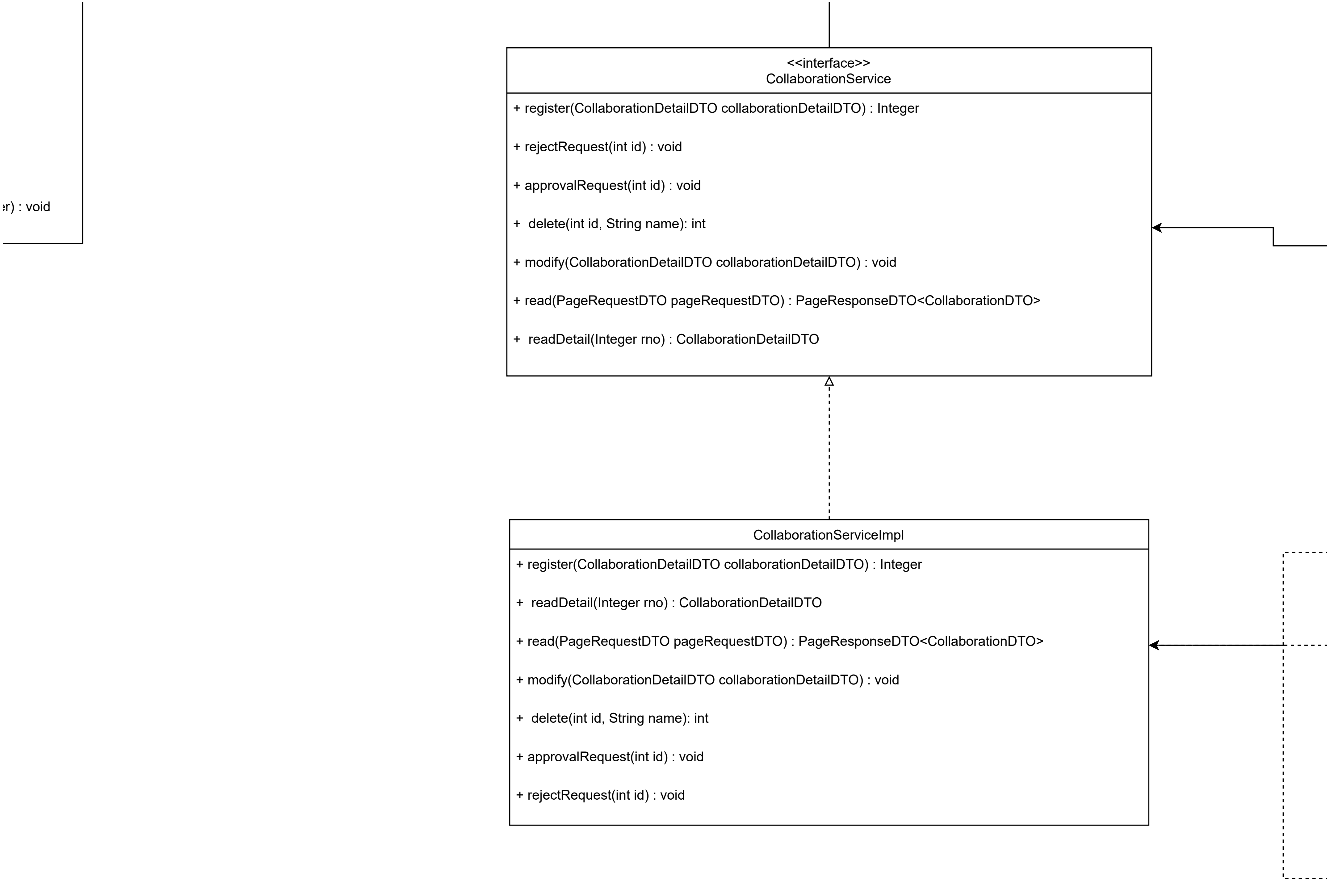
+ modifyPackage(PackageDTO packageDTO) : void

+ viewPackage(int project_id) : List<PackageDTO>

+ viewReward(int package_id) : List<RewardDTO>

+ viewOption(int reward_id) : List<OptionDTO>

+registerPackage(PackageDTO packageDTO,List<RewardDTO> rewardDTOList, Integer project_id) : void



r) : void

<<interface>>
CollaborationService

+ register(CollaborationDetailDTO collaborationDetailDTO) : Integer

+ rejectRequest(int id) : void

+ approvalRequest(int id) : void

+ delete(int id, String name): int

+ modify(CollaborationDetailDTO collaborationDetailDTO) : void

+ read(PageRequestDTO pageRequestDTO) : PageResponseDTO<CollaborationDTO>

+ readDetail(Integer rno) : CollaborationDetailDTO

CollaborationServiceImpl

+ register(CollaborationDetailDTO collaborationDetailDTO) : Integer

+ readDetail(Integer rno) : CollaborationDetailDTO

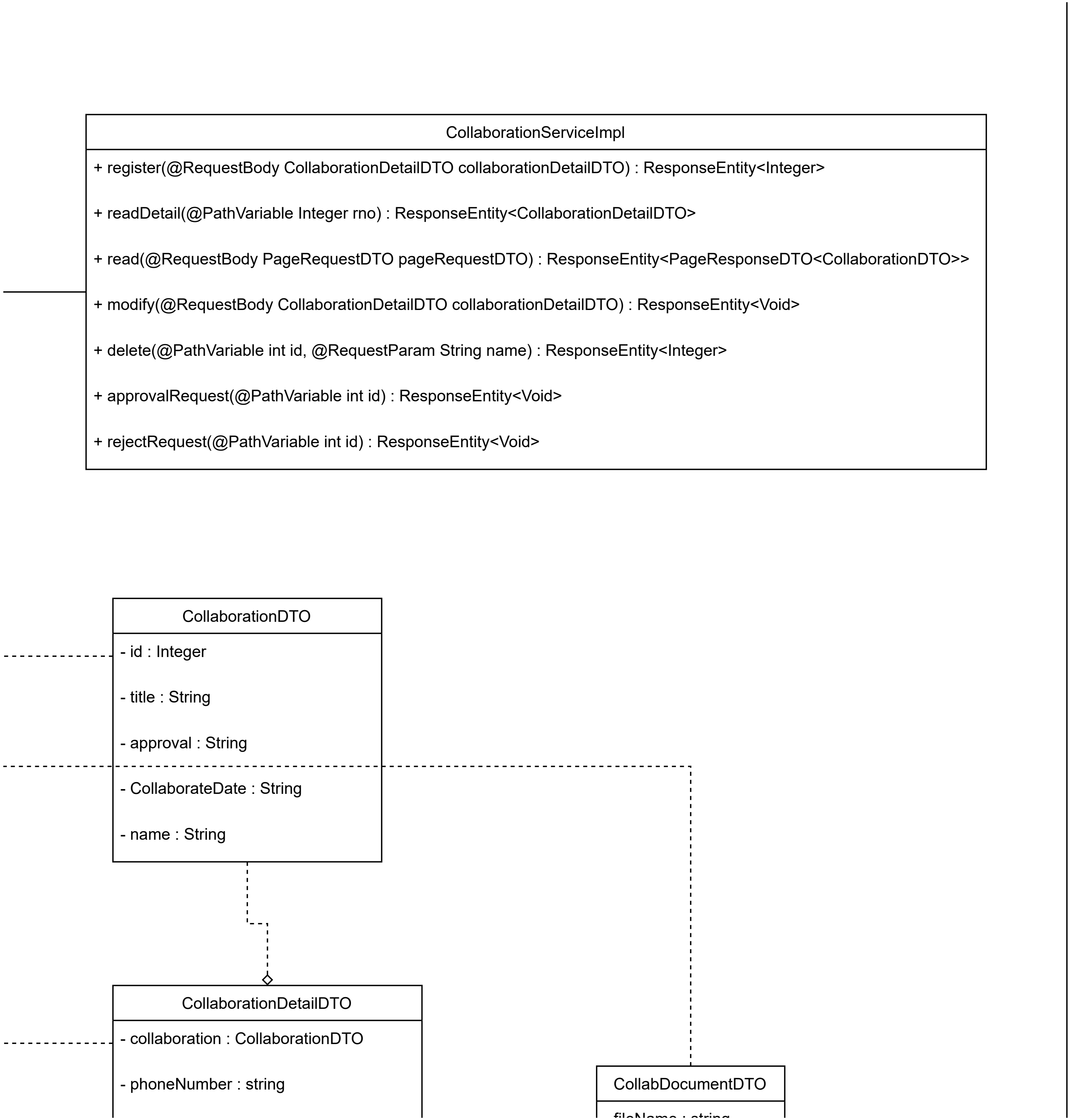
+ read(PageRequestDTO pageRequestDTO) : PageResponseDTO<CollaborationDTO>

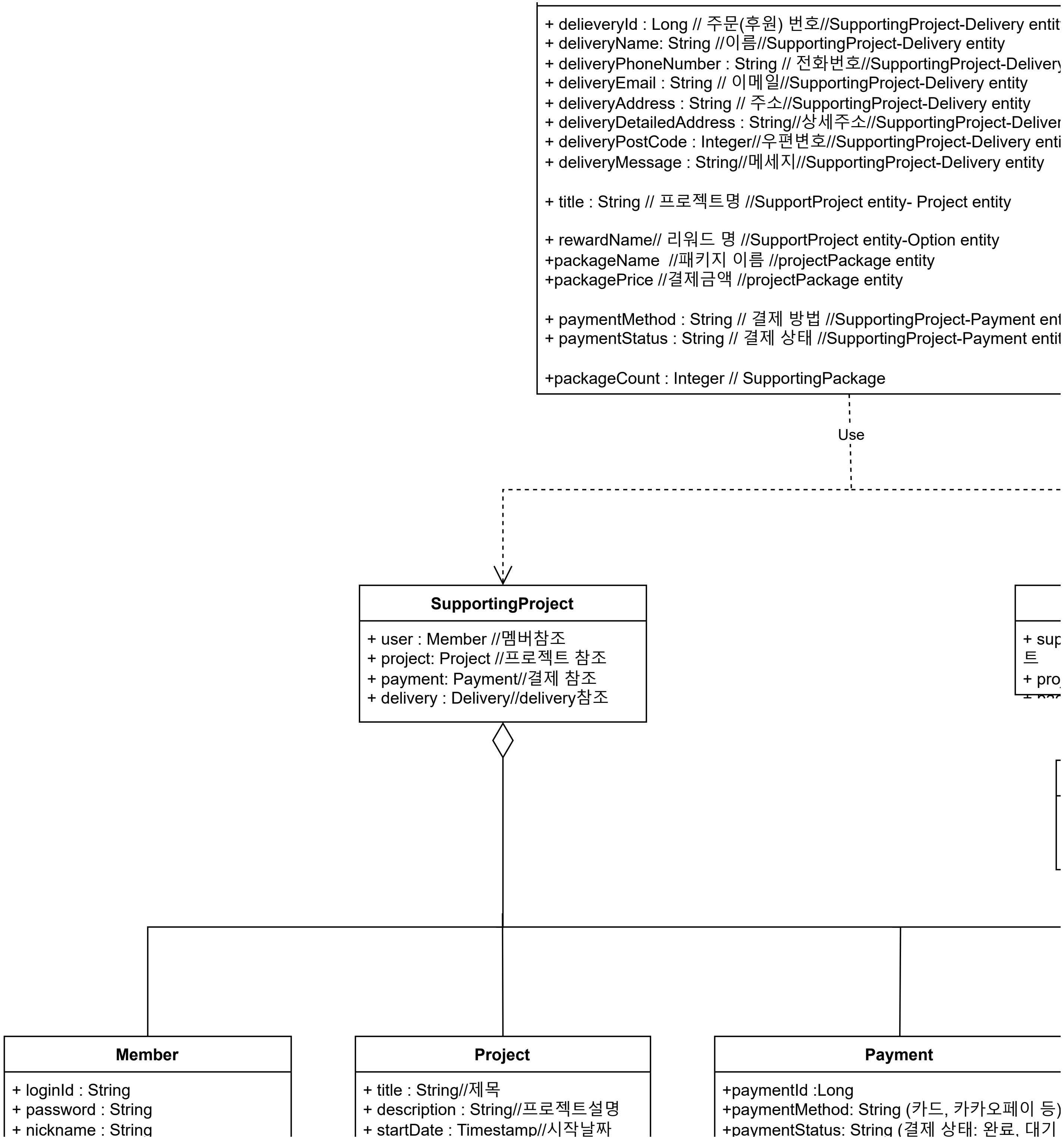
+ modify(CollaborationDetailDTO collaborationDetailDTO) : void

+ delete(int id, String name): int

+ approvalRequest(int id) : void

+ rejectRequest(int id) : void


CollaborationServiceImpl
+ register(@RequestBody CollaborationDetailDTO collaborationDetailDTO) : ResponseEntity<Integer>
+ readDetail(@PathVariable Integer rno) : ResponseEntity<CollaborationDetailDTO>
+ read(@RequestBody PageRequestDTO pageRequestDTO) : ResponseEntity<PageResponseDTO<CollaborationDTO>>
+ modify(@RequestBody CollaborationDetailDTO collaborationDetailDTO) : ResponseEntity<Void>
+ delete(@PathVariable int id, @RequestParam String name) : ResponseEntity<Integer>
+ approvalRequest(@PathVariable int id) : ResponseEntity<Void>
+ rejectRequest(@PathVariable int id) : ResponseEntity<Void>
CollaborationDTO
- id : Integer
- title : String
- approval : String
- CollaborateDate : String
- name : String
CollaborationDetailDTO
- collaboration : CollaborationDTO
- phoneNumber : string
CollabDocumentDTO
fileName : string



+ delieveryId : Long // 주문(후원) 번호//SupportingProject-Delivery entit
+ deliveryName: String //이름//SupportingProject-Delivery entity
+ deliveryPhoneNumber : String // 전화번호//SupportingProject-Delivery
+ deliveryEmail : String // 이메일//SupportingProject-Delivery entity
+ deliveryAddress : String // 주소//SupportingProject-Delivery entity
+ deliveryDetailedAddress : String//상세주소//SupportingProject-Deliver
+ deliveryPostCode : Integer//우편번호//SupportingProject-Delivery enti
+ deliveryMessage : String//메세지//SupportingProject-Delivery entity

+ title : String // 프로젝트명 //SupportProject entity- Project entity

+ rewardName// 리워드 명 //SupportProject entity-Option entity
+packageName //패키지 이름 //projectPackage entity
+packagePrice //결제금액 //projectPackage entity

+ paymentMethod : String // 결제 방법 //SupportingProject-Payment ent
+ paymentStatus : String // 결제 상태 //SupportingProject-Payment entit

+packageCount : Integer // SupportingPackage

SupportingProject

+ user : Member //멤버참조
+ project: Project //프로젝트 참조
+ payment: Payment//결제 참조
+ delivery : Delivery//delivery참조

+ sup
트
+ pro

Member

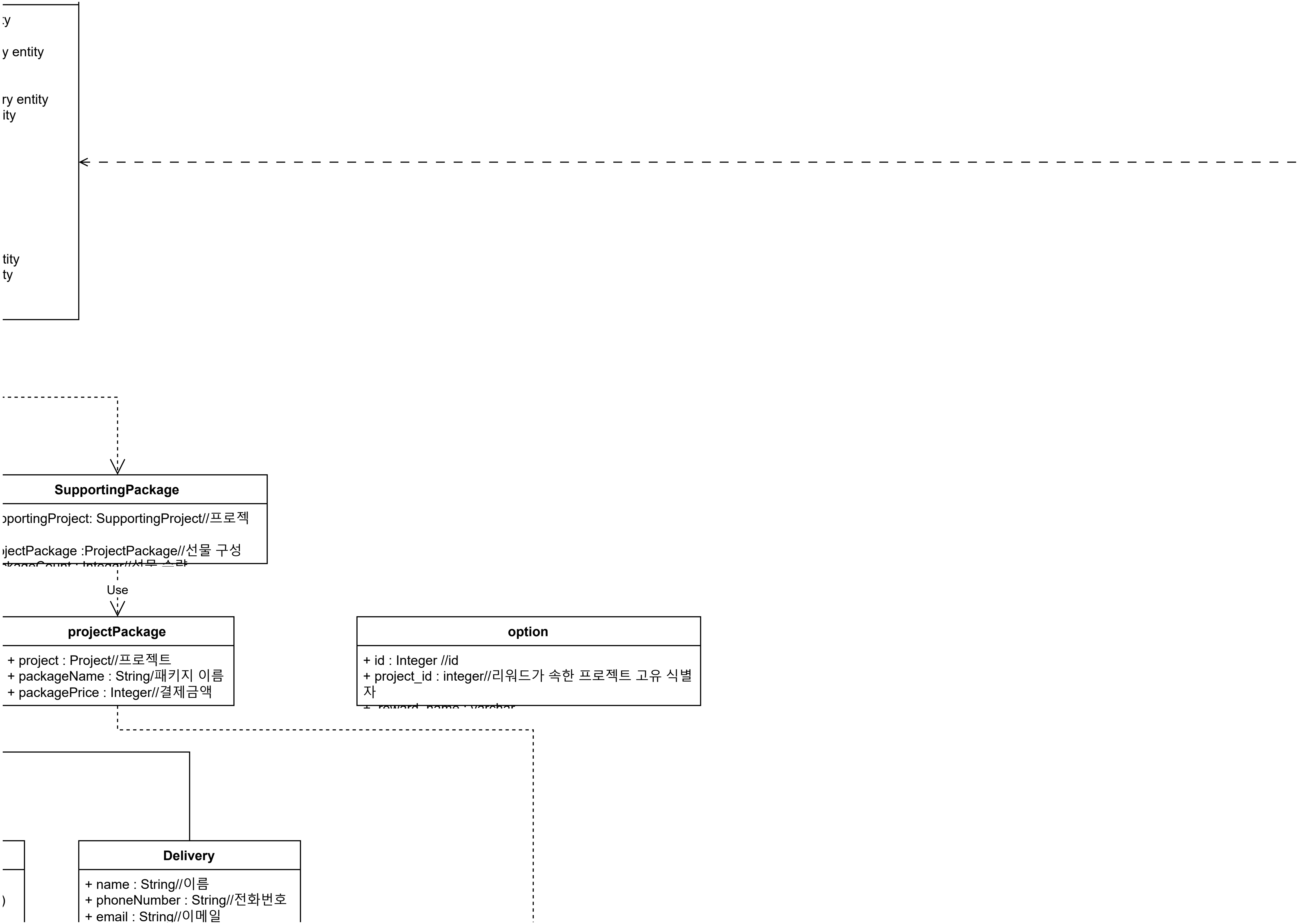
+ loginId : String
+ password : String
+ nickname : String

Project

+ title : String//제목
+ description : String//프로젝트설명
+ startDate : Timestamp//시작날짜

Payment

+paymentId :Long
+paymentMethod: String (카드, 카카오페이 등)
+paymentStatus: String (결제 상태: 완료, 대기



ty

y entity

ry entity
ity

tity
ty

SupportingPackage

pportingProject: SupportingProject//프로젝

jectPackage :ProjectPackage//선물 구성
kageCount : Integer//선물 수량

Use

projectPackage

+ project : Project//프로젝트
+ packageName : String/패키지 이름
+ packagePrice : Integer//결제금액

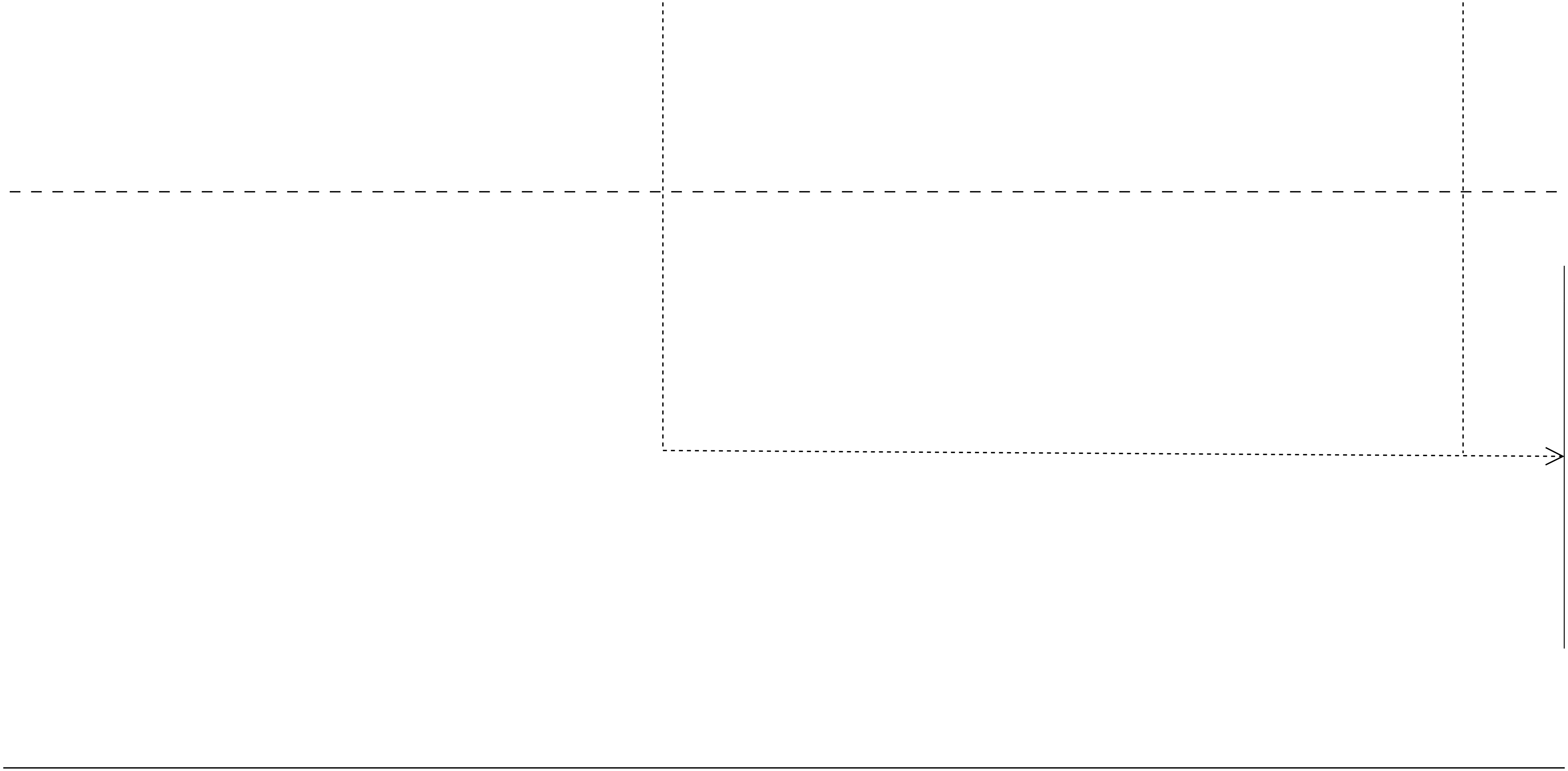
option

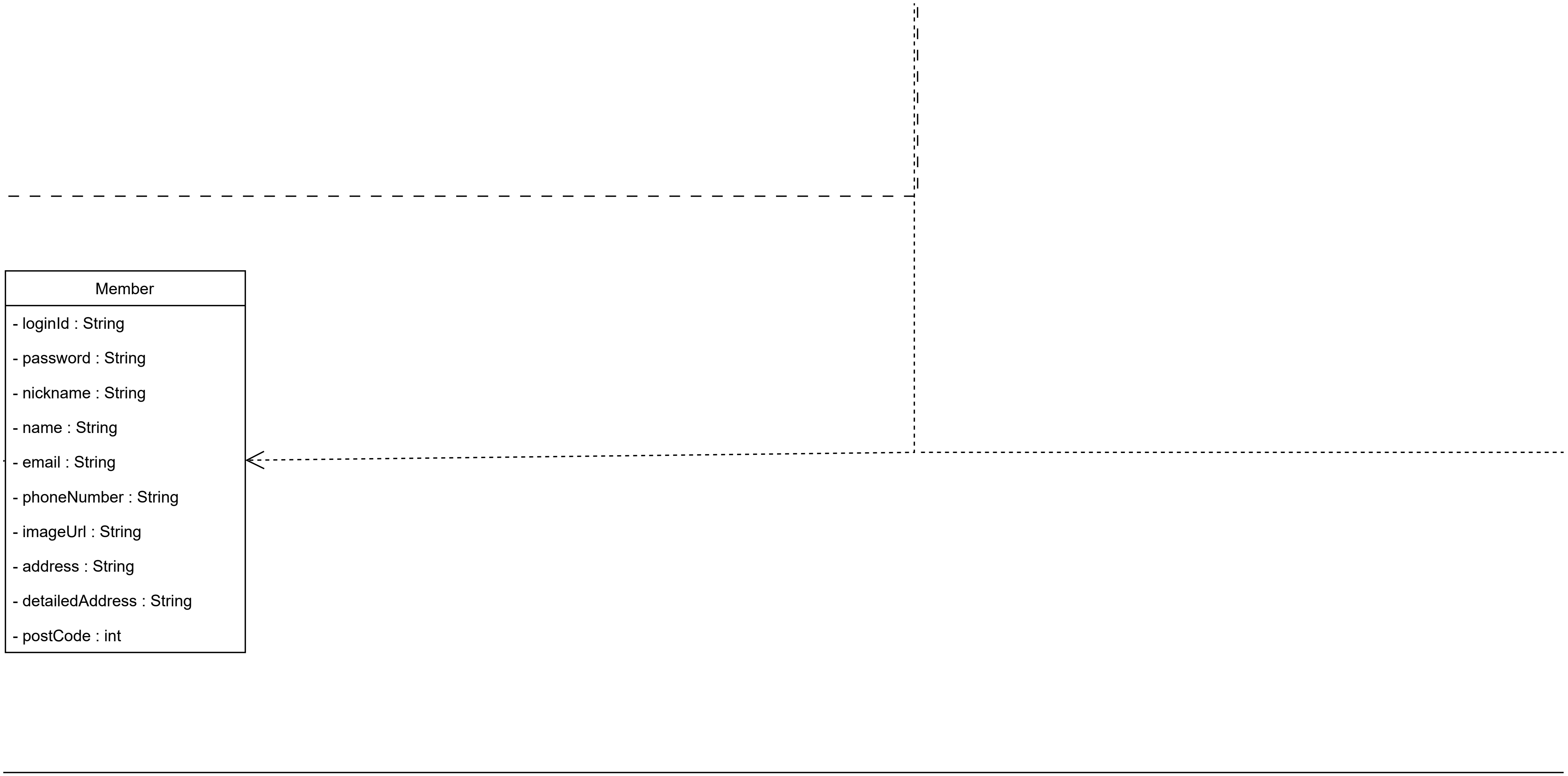
+ id : Integer //id
+ project_id : integer//리워드가 속한 프로젝트 고유 식별자
+ reward_name : varchar

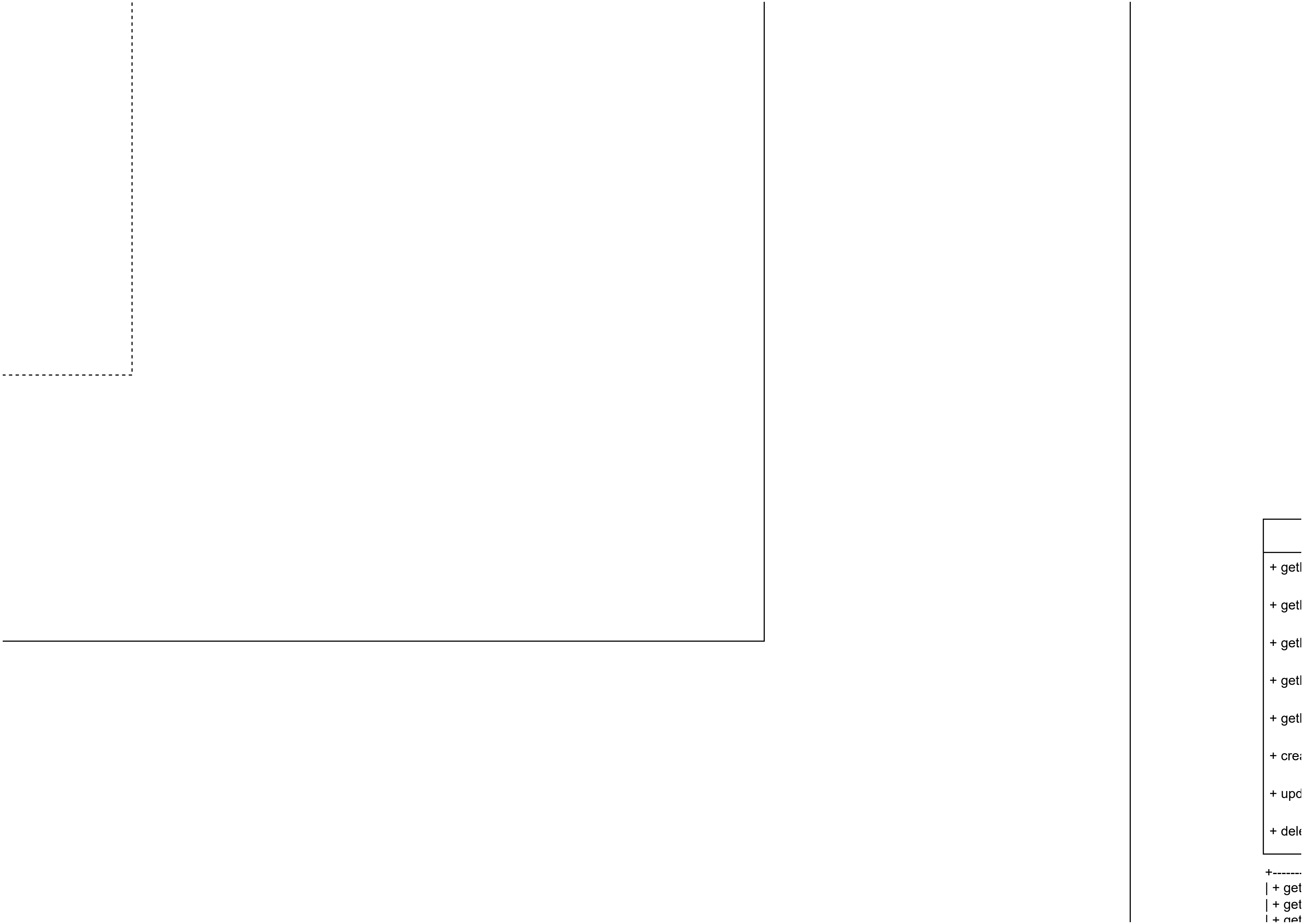
)

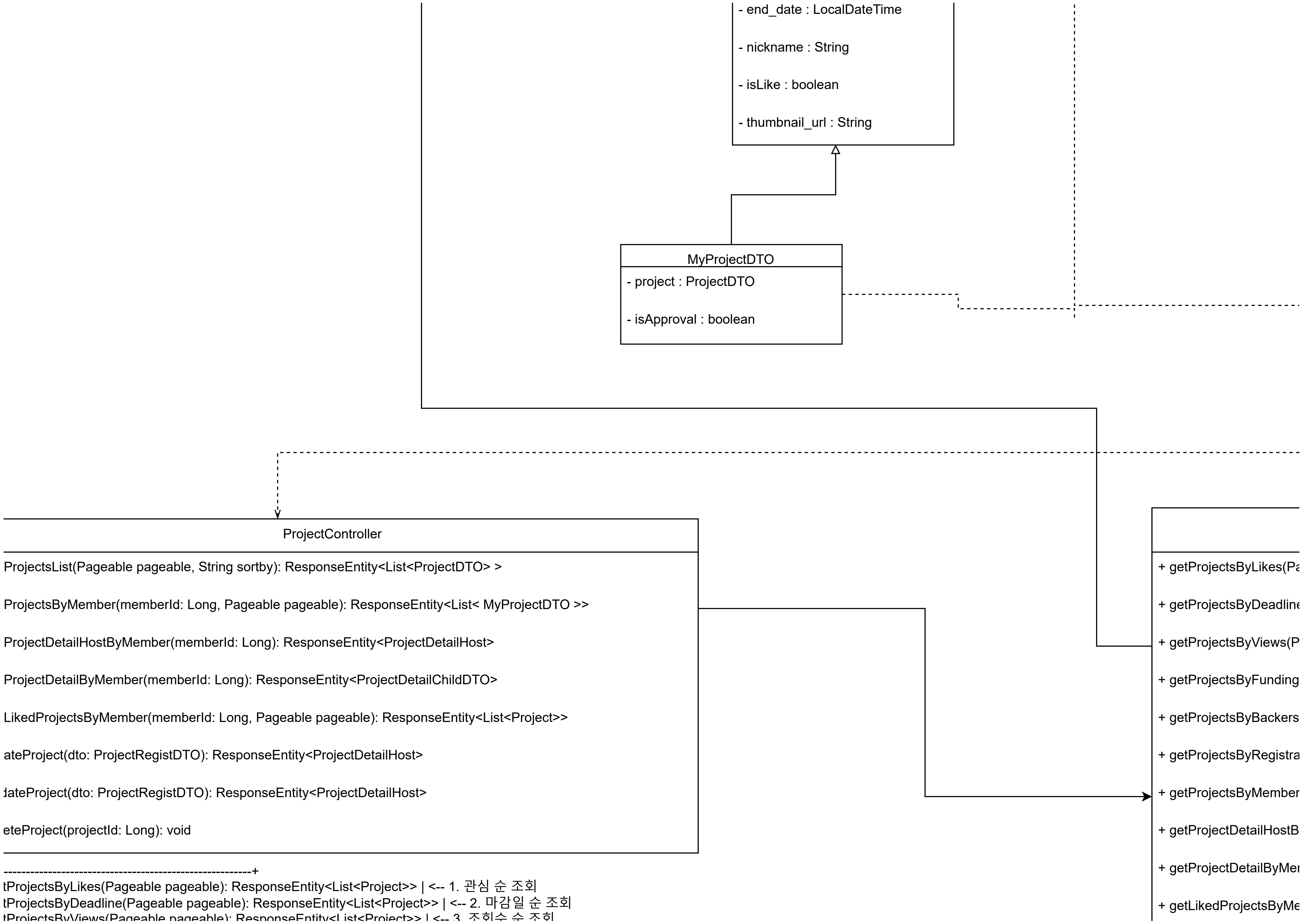
Delivery

+ name : String//이름
+ phoneNumber : String//전화번호
+ email : String//이메일




Member
- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- imageUrl : String
- address : String
- detailedAddress : String
- postCode : int


+ getI
+ getI
+ getI
+ getI
+ getI
+ crea
+ upd
+ dele
+-------
| + get
| + get
| + get



- end_date : LocalDateTime

- nickname : String

- isLike : boolean

- thumbnail_url : String

MyProjectDTO

- project : ProjectDTO

- isApproval : boolean

ProjectController

ProjectsList(Pageable pageable, String sortby): ResponseEntity<List<ProjectDTO> >

ProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List< MyProjectDTO >>

ProjectDetailHostByMember(memberId: Long): ResponseEntity<ProjectDetailHost>

ProjectDetailByMember(memberId: Long): ResponseEntity<ProjectDetailChildDTO>

LikedProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>>

ateProject(dto: ProjectRegistDTO): ResponseEntity<ProjectDetailHost>

dateProject(dto: ProjectRegistDTO): ResponseEntity<ProjectDetailHost>

eteProject(projectId: Long): void

```
---------------------------------------------------------+
tProjectsByLikes(Pageable pageable): ResponseEntity<List<Project>> | <-- 1. 관심 순 조회
tProjectsByDeadline(Pageable pageable): ResponseEntity<List<Project>> | <-- 2. 마감일 순 조회
tProjectsByViews(Pageable pageable): ResponseEntity<List<Project>> | <-- 3. 조회수 순 조회
```

+ getProjectsByLikes(Pa

+ getProjectsByDeadline

+ getProjectsByViews(P

+ getProjectsByFunding

+ getProjectsByBackers

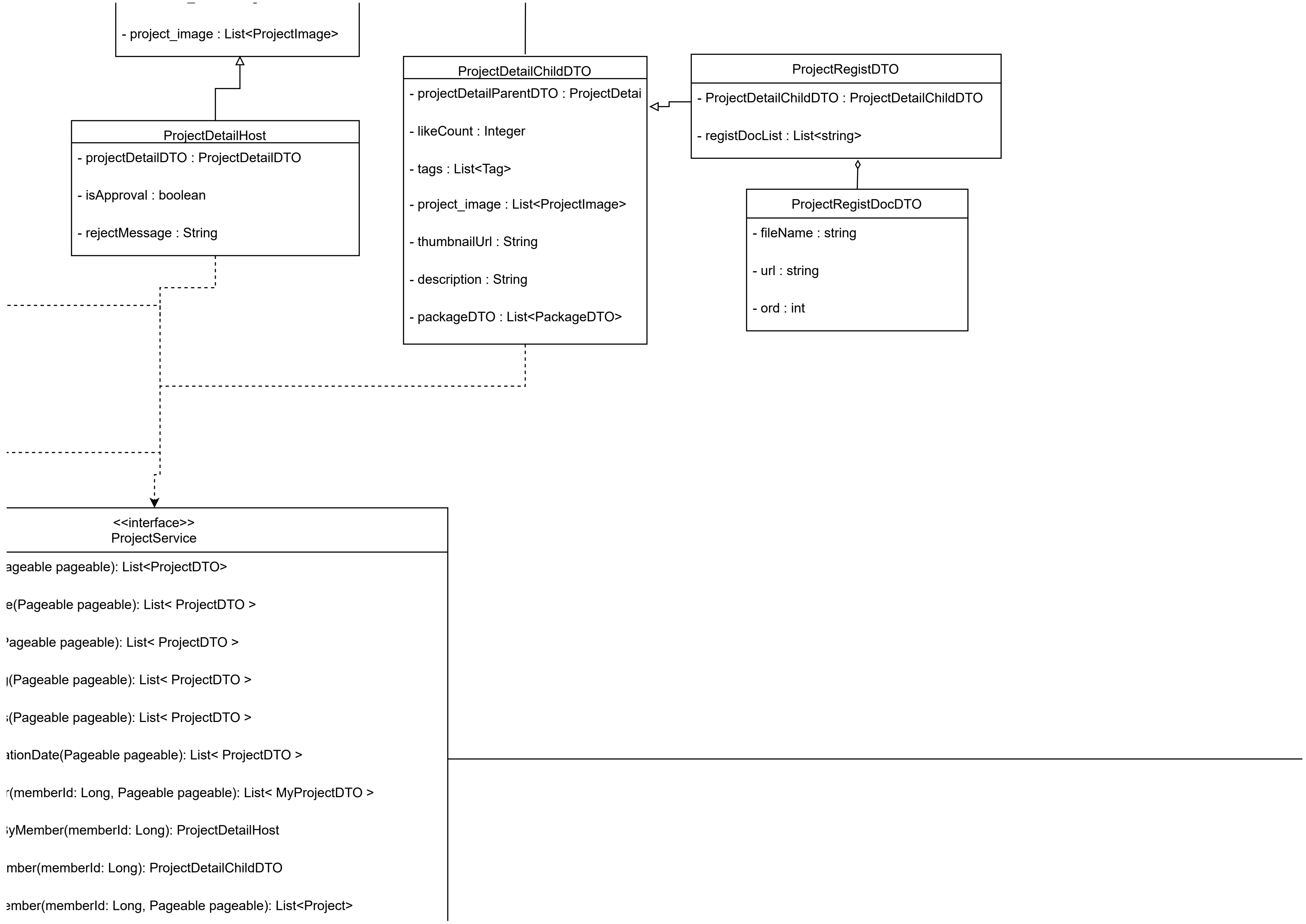
+ getProjectsByRegistra

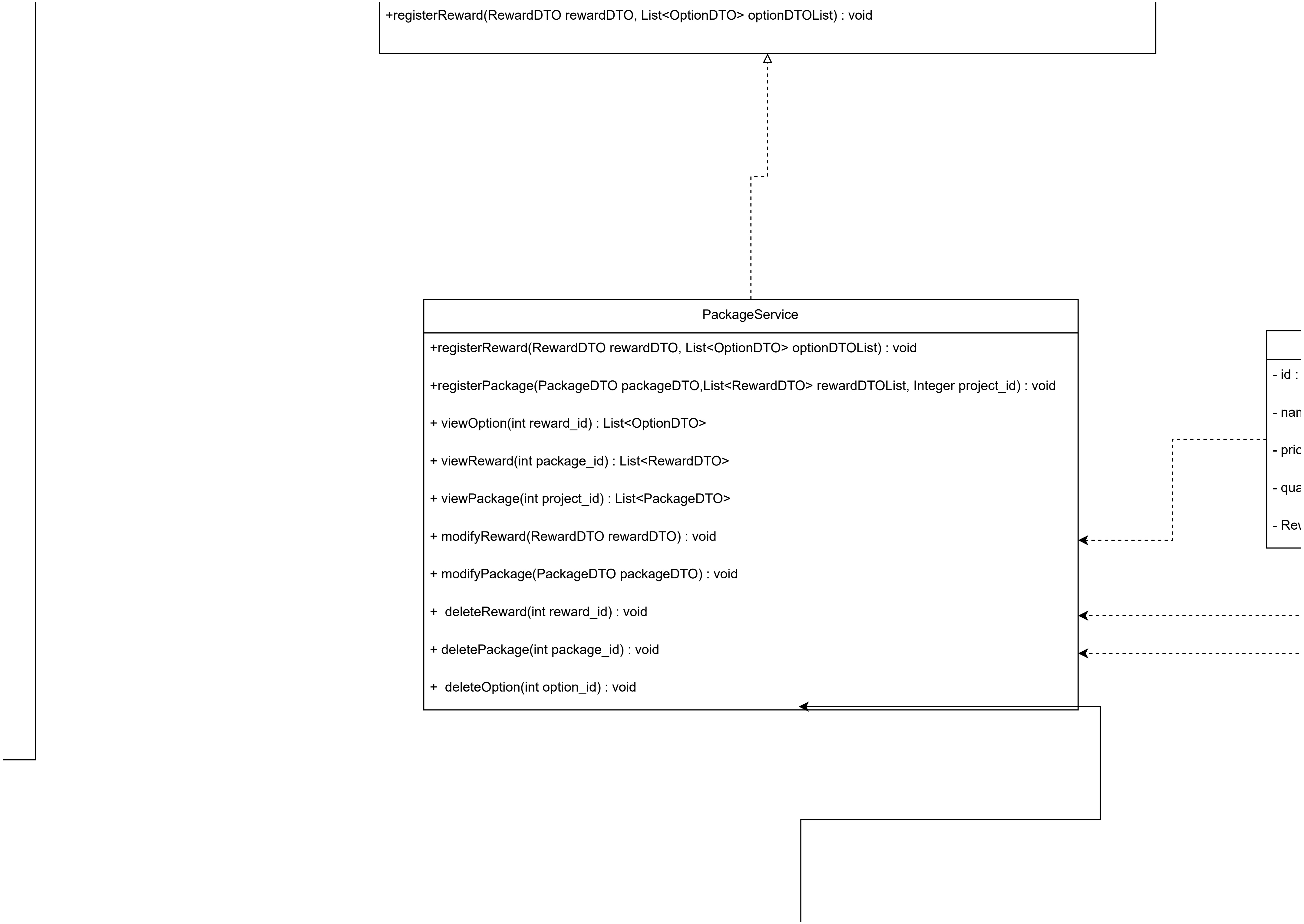
+ getProjectsByMember

+ getProjectDetailHostB

+ getProjectDetailByMer

+ getLikedProjectsByMe


- project_image : List<ProjectImage>
ProjectDetailHost
- projectDetailDTO : ProjectDetailDTO
- isApproval : boolean
- rejectMessage : String
ProjectDetailChildDTO
- projectDetailParentDTO : ProjectDetai
- likeCount : Integer
- tags : List<Tag>
- project_image : List<ProjectImage>
- thumbnailUrl : String
- description : String
- packageDTO : List<PackageDTO>
ProjectRegistDTO
- ProjectDetailChildDTO : ProjectDetailChildDTO
- registDocList : List<string>
ProjectRegistDocDTO
- fileName : string
- url : string
- ord : int
<<interface>>
ProjectService
ageable pageable): List<ProjectDTO>
e(Pageable pageable): List< ProjectDTO >
'ageable pageable): List< ProjectDTO >
ı(Pageable pageable): List< ProjectDTO >
;(Pageable pageable): List< ProjectDTO >
ationDate(Pageable pageable): List< ProjectDTO >
r(memberId: Long, Pageable pageable): List< MyProjectDTO >
;yMember(memberId: Long): ProjectDetailHost
mber(memberId: Long): ProjectDetailChildDTO
ember(memberId: Long, Pageable pageable): List<Project>



+registerReward(RewardDTO rewardDTO, List<OptionDTO> optionDTOList) : void

PackageService

+registerReward(RewardDTO rewardDTO, List<OptionDTO> optionDTOList) : void

+registerPackage(PackageDTO packageDTO,List<RewardDTO> rewardDTOList, Integer project_id) : void

+ viewOption(int reward_id) : List<OptionDTO>

+ viewReward(int package_id) : List<RewardDTO>

+ viewPackage(int project_id) : List<PackageDTO>

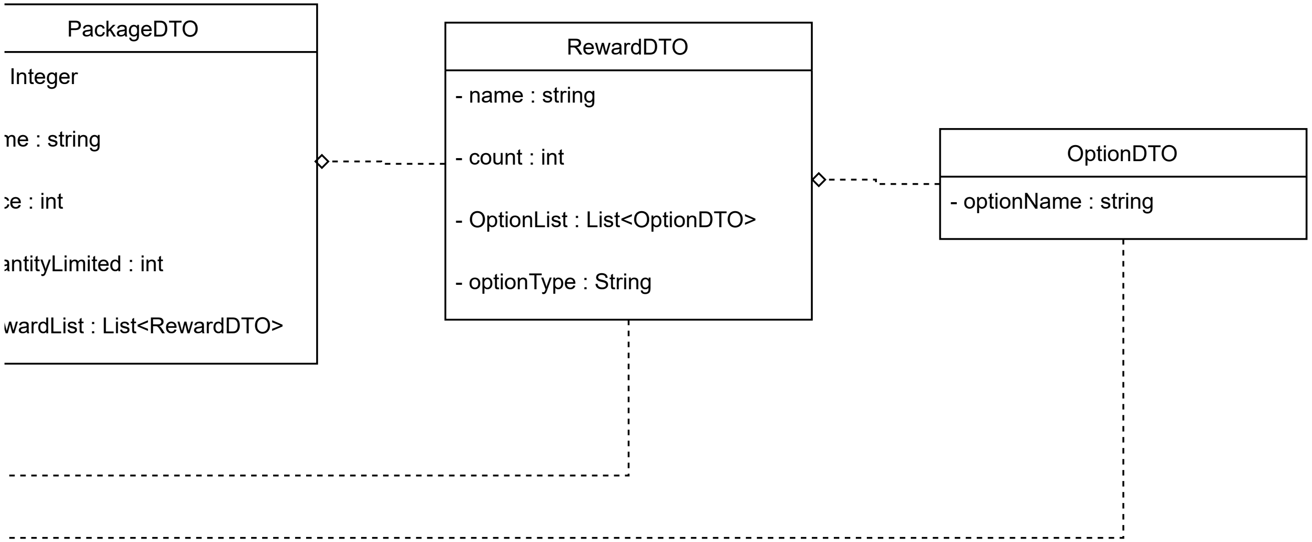
+ modifyReward(RewardDTO rewardDTO) : void

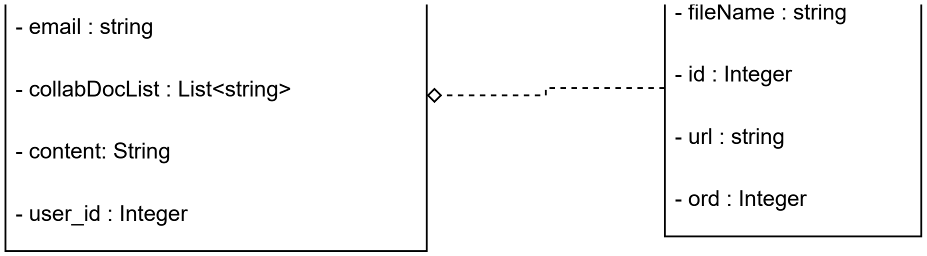
+ modifyPackage(PackageDTO packageDTO) : void

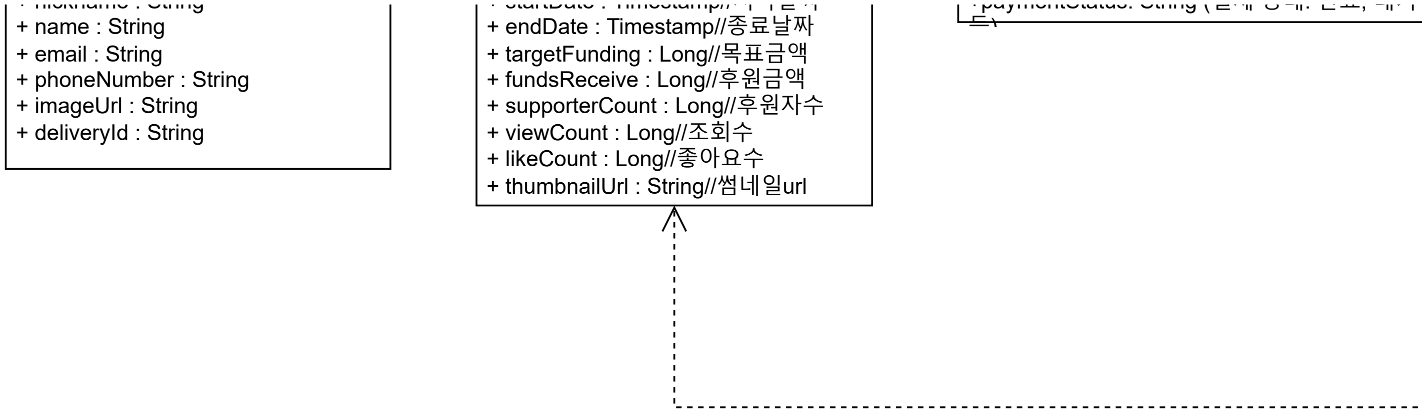
+ deleteReward(int reward_id) : void

+ deletePackage(int package_id) : void

+ deleteOption(int option_id) : void


PackageDTO
Integer
ne : string
ce : int
antityLimited : int
wardList : List<RewardDTO>
RewardDTO
- name : string
- count : int
- OptionList : List<OptionDTO>
- optionType : String
OptionDTO
- optionName : string

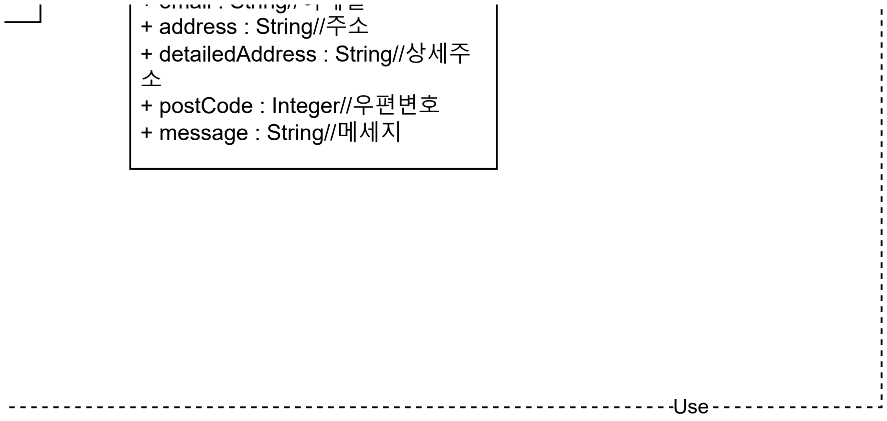

- email : string
- collabDocList : List<string>
- content: String
- user_id : Integer
- fileName : string
- id : Integer
- url : string
- ord : Integer



+ nickname : String
+ name : String
+ email : String
+ phoneNumber : String
+ imageUrl : String
+ deliveryId : String

+ startDate : Timestamp//시작날짜
+ endDate : Timestamp//종료날짜
+ targetFunding : Long//목표금액
+ fundsReceive : Long//후원금액
+ supporterCount : Long//후원자수
+ viewCount : Long//조회수
+ likeCount : Long//좋아요수
+ thumbnailUrl : String//썸네일url

+paymentStatus: String (결제 상태: 완료, 대기 등)



+ email : String//이메일
+ address : String//주소
+ detailedAddress : String//상세주소
+ postCode : Integer//우편변호
+ message : String//메세지

```
| + get
| + get
| + get
| + get
| + get
| + get
| + get
| + cre
| + upd
| + del
+-------
```

```
tProjectsByViews(Pageable pageable): ResponseEntity<List<Project>> | <-- 3. 조회수 순 조회
tProjectsByFunding(Pageable pageable): ResponseEntity<List<Project>> | <-- 4. 후원금 순 조회
tProjectsByBackers(Pageable pageable): ResponseEntity<List<Project>> | <-- 5. 후원자 순 조회
tProjectsByRegistrationDate(Pageable pageable): ResponseEntity<List<Project>> | <-- 6. 등록일 순 조회
tProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>> | <-- 7. 특정 회원의 프로젝트 조회
tProjectDetailByMember(memberId: Long): ResponseEntity<ProjectDetailDTO> | <-- 8. 특정 회원의 프로젝트 상세 보기
tLikedProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>> | <-- 9. 특정 회원의 관심 프로젝트 조회
eateProject(dto: ProjectDTO): ResponseEntity<Project> | <-- 10. 프로젝트 생성
dateProject(dto: ProjectDTO): ResponseEntity<Project> | <-- 11. 프로젝트 수정
leteProject(projectId: Long): ResponseEntity<Void> | <-- 12. 프로젝트 삭제
---------------------------------------------------------+
```

```
+ createProject(dto: Pro

+ updateProject(dto: Pro

+ deleteProject(projectId
```

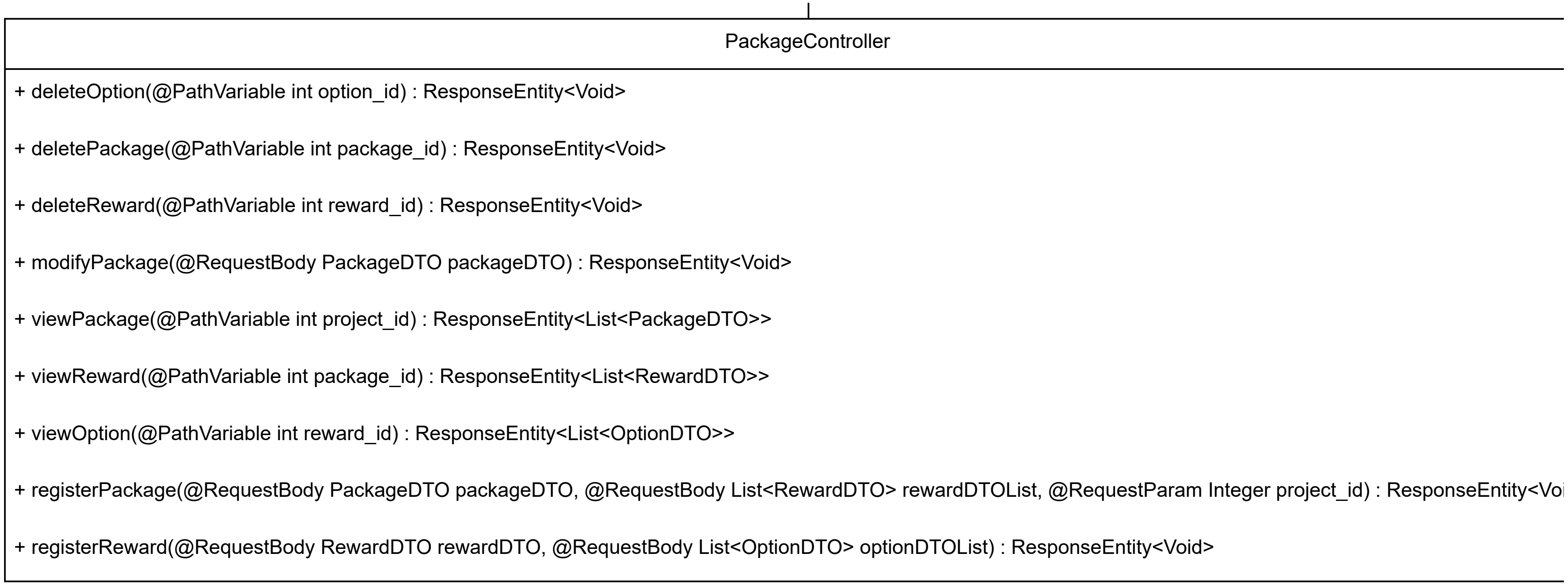
이건 그냥 메모... 뭔지 모를까봐요...

```
| + getProjectsByLikes(Pageable pag
| + getProjectsByDeadline(Pageable
| + getProjectsByViews(Pageable pa
| + getProjectsByFunding(Pageable
| + getProjectsByBackers(Pageable
| + getProjectsByRegistrationDate(P
| + getProjectsByMember(memberId
| + getProjectDetailByMember(meml
| + getLikedProjectsByMember(mem
| + createProject(dto: ProjectDTO): F
| + updateProject(dto: ProjectDTO):
| + deleteProject(projectId: Long): vo
jectRegistDTO): ProjectDetailHost

ojectRegistDTO): ProjectDetailHost

d: Long): void
```

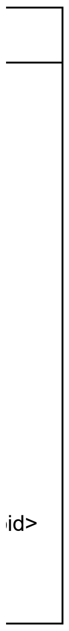
ㅎㅎ

```
geable): List<Project>     |  <-- 1. 관심표시가 많은 순으로 프로젝트 조회 (페이지네이션)
e pageable): List<Project>  |  <-- 2. 마감일에 가까운 순으로 프로젝트 조회 (페이지네이션)
ageable): List<Project>     |  <-- 3. 조회수가 많은 순으로 프로젝트 조회 (페이지네이션)
 pageable): List<Project>   |  <-- 4. 후원금액이 많은 순으로 프로젝트 조회 (페이지네이션)
 pageable): List<Project>   |  <-- 5. 후원자 수가 많은 순으로 프로젝트 조회 (페이지네이션)
Pageable pageable): List<Project> |  <-- 6. 등록일 순으로 프로젝트 조회 (페이지네이션)
d: Long, Pageable pageable): List<Project> |  <-- 7. 특정 회원의 프로젝트 조회 (memberId)
berId: Long): ProjectDetailDTO |  <-- 8. 특정 회원의 프로젝트 상세 보기 (memberId)
nberId: Long, Pageable pageable): List<Project> |  <-- 9. 특정 회원의 관심 프로젝트 조회 (memberId)
Project                  |  <-- 10. 프로젝트 생성 (dto)
 Project                 |  <-- 11. 프로젝트 수정 (dto)
oid                     |  <-- 12. 프로젝트 삭제 (projectId
```

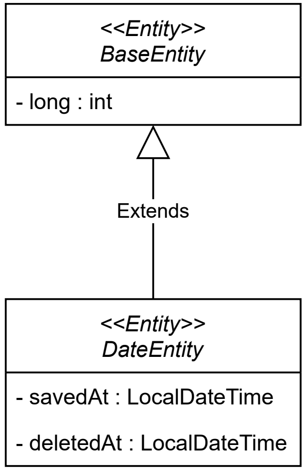
따로 가져가지는 않고 여기서 작업할게용!

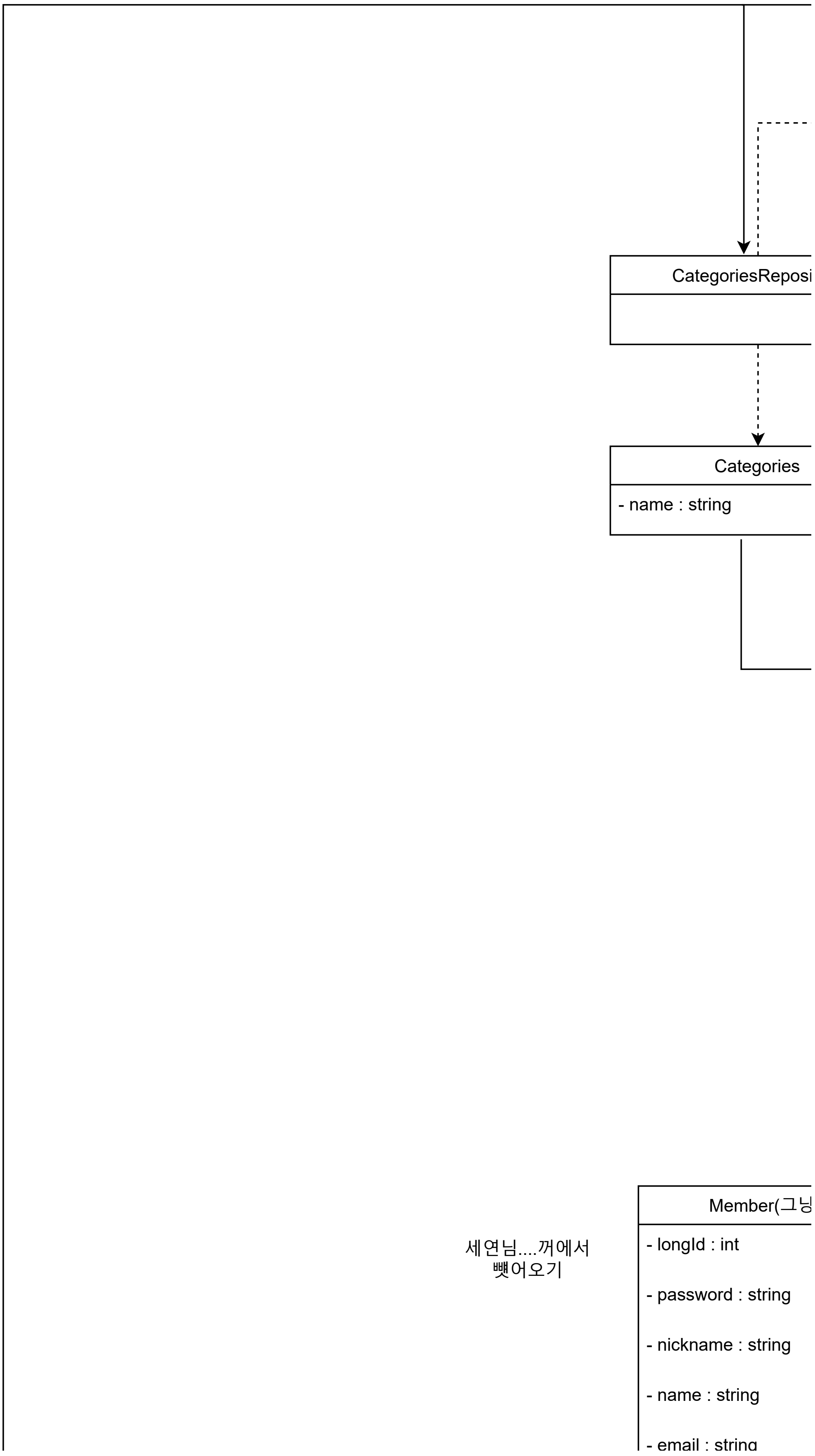


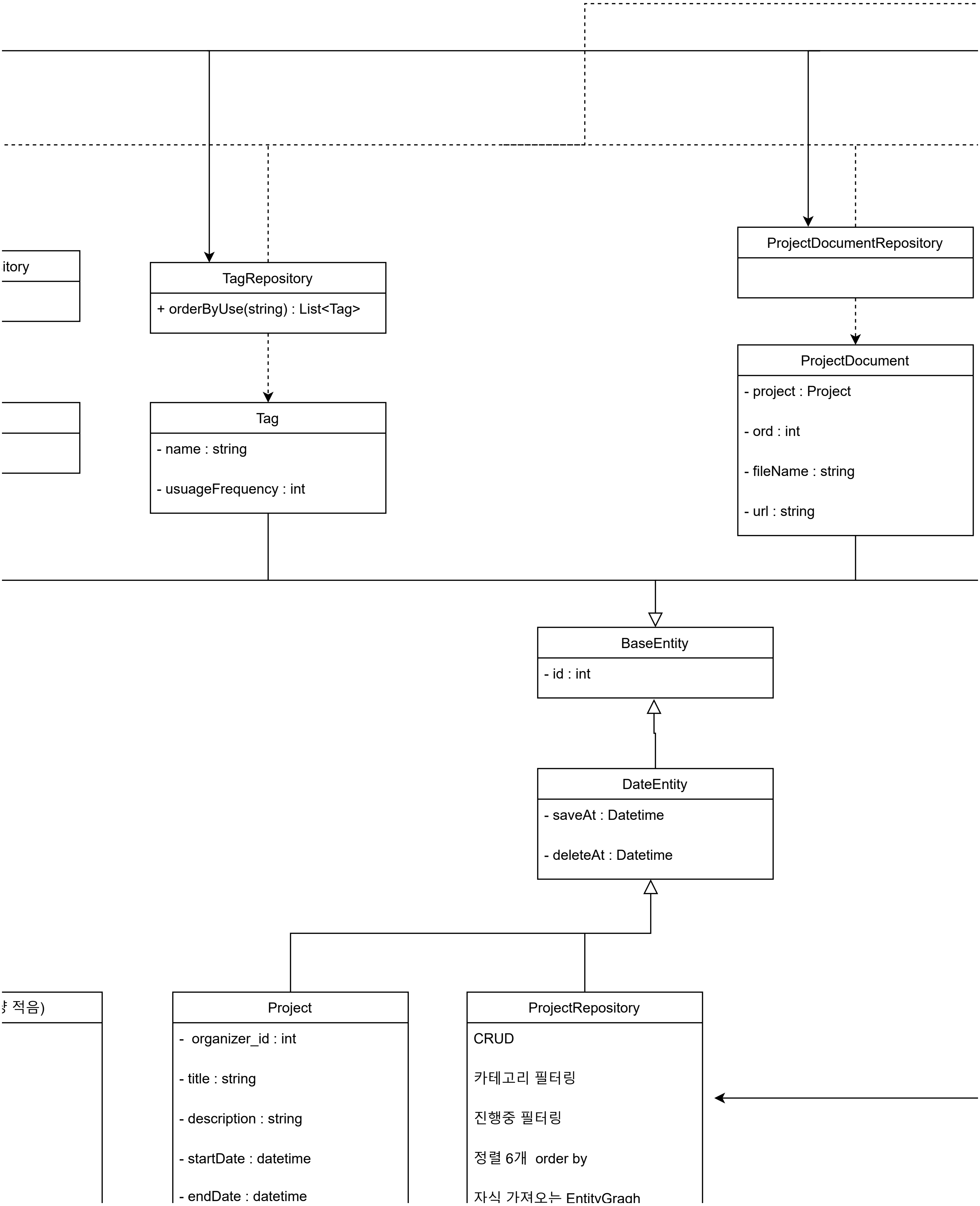
```
PackageController
+ deleteOption(@PathVariable int option_id) : ResponseEntity<Void>
+ deletePackage(@PathVariable int package_id) : ResponseEntity<Void>
+ deleteReward(@PathVariable int reward_id) : ResponseEntity<Void>
+ modifyPackage(@RequestBody PackageDTO packageDTO) : ResponseEntity<Void>
+ viewPackage(@PathVariable int project_id) : ResponseEntity<List<PackageDTO>>
+ viewReward(@PathVariable int package_id) : ResponseEntity<List<RewardDTO>>
+ viewOption(@PathVariable int reward_id) : ResponseEntity<List<OptionDTO>>
+ registerPackage(@RequestBody PackageDTO packageDTO, @RequestBody List<RewardDTO> rewardDTOList, @RequestParam Integer project_id) : ResponseEntity<Voi
+ registerReward(@RequestBody RewardDTO rewardDTO, @RequestBody List<OptionDTO> optionDTOList) : ResponseEntity<Void>
```


id>

common


<<Entity>>
BaseEntity
- long : int
Extends
<<Entity>>
DateEntity
- savedAt : LocalDateTime
- deletedAt : LocalDateTime


CategoriesReposi
Categories
- name : string
세연님....꺼에서
뺏어오기
Member(그닝
- longId : int
- password : string
- nickname : string
- name : string
- email : string



itory

TagRepository

+ orderByUse(string) : List<Tag>

Tag

- name : string

- usuageFrequency : int

ProjectDocumentRepository

ProjectDocument

- project : Project

- ord : int

- fileName : string

- url : string

BaseEntity

- id : int

DateEntity

- saveAt : Datetime

- deleteAt : Datetime

; 적음)

Project

- organizer_id : int

- title : string

- description : string

- startDate : datetime

- endDate : datetime

ProjectRepository

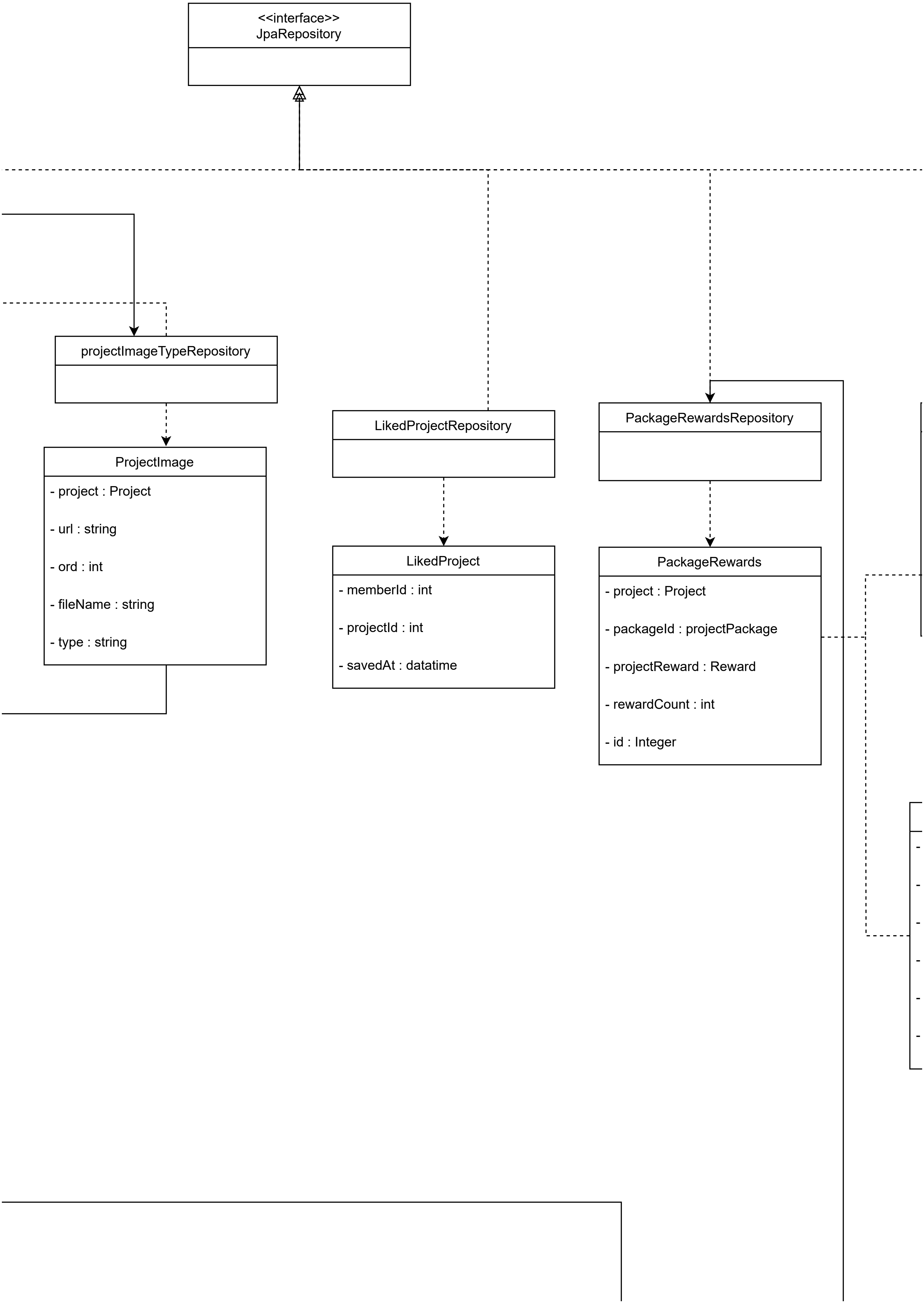
CRUD

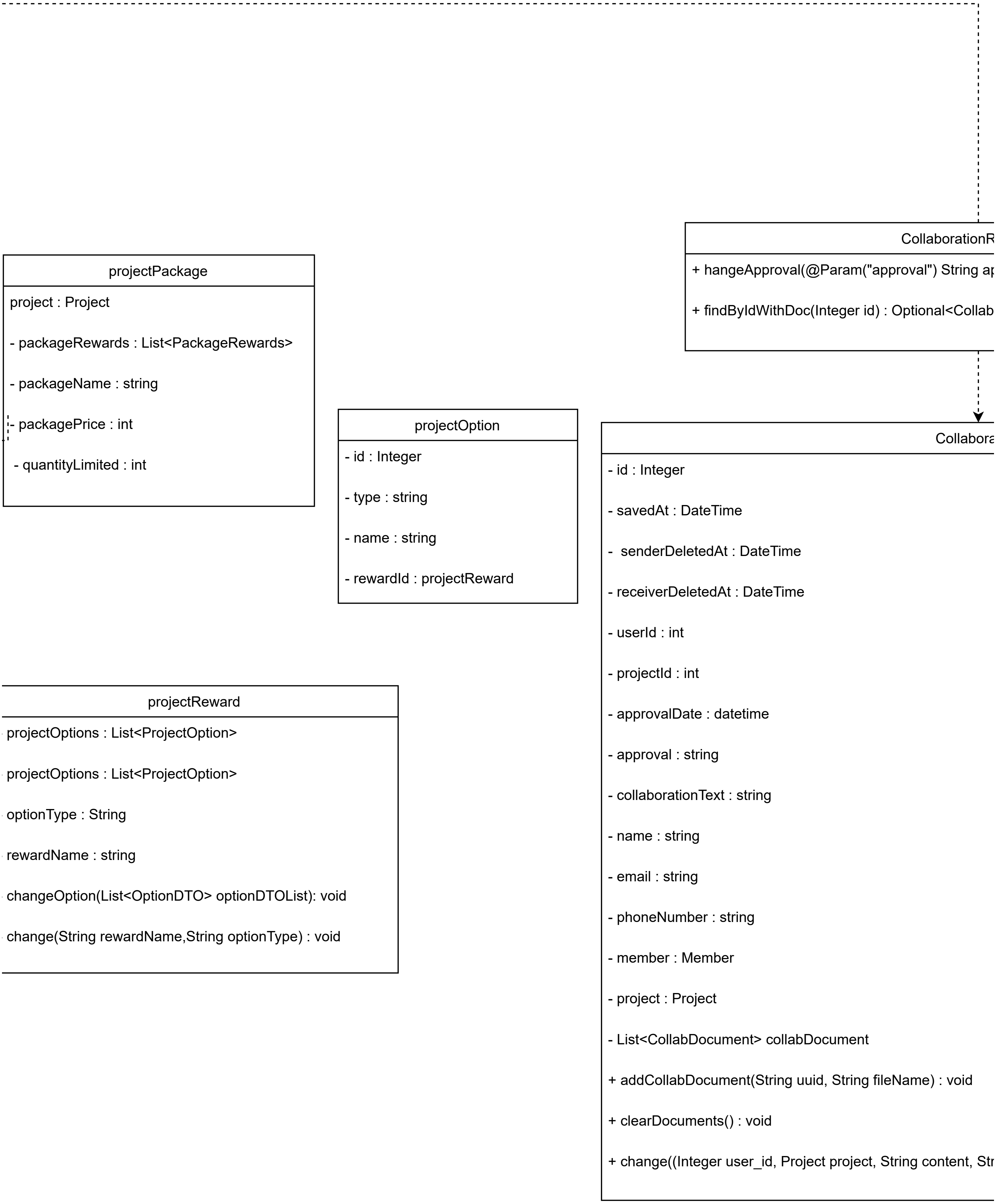
카테고리 필터링

진행중 필터링

정렬 6개 order by

자식 가져오는 EntityGragh


<<interface>>
JpaRepository
projectImageTypeRepository
ProjectImage
- project : Project
- url : string
- ord : int
- fileName : string
- type : string
LikedProjectRepository
LikedProject
- memberId : int
- projectId : int
- savedAt : datatime
PackageRewardsRepository
PackageRewards
- project : Project
- packageId : projectPackage
- projectReward : Reward
- rewardCount : int
- id : Integer
-
-
-
-
-
-



CollaborationR

+ hangeApproval(@Param("approval") String ap

+ findByIdWithDoc(Integer id) : Optional<Collab

projectPackage

project : Project

- packageRewards : List<PackageRewards>

- packageName : string

- packagePrice : int

- quantityLimited : int

projectOption

- id : Integer

- type : string

- name : string

- rewardId : projectReward

Collabora

- id : Integer

- savedAt : DateTime

- senderDeletedAt : DateTime

- receiverDeletedAt : DateTime

- userId : int

- projectId : int

- approvalDate : datetime

- approval : string

- collaborationText : string

- name : string

- email : string

- phoneNumber : string

- member : Member

- project : Project

- List<CollabDocument> collabDocument

+ addCollabDocument(String uuid, String fileName) : void

+ clearDocuments() : void

+ change((Integer user_id, Project project, String content, Str

projectReward

projectOptions : List<ProjectOption>

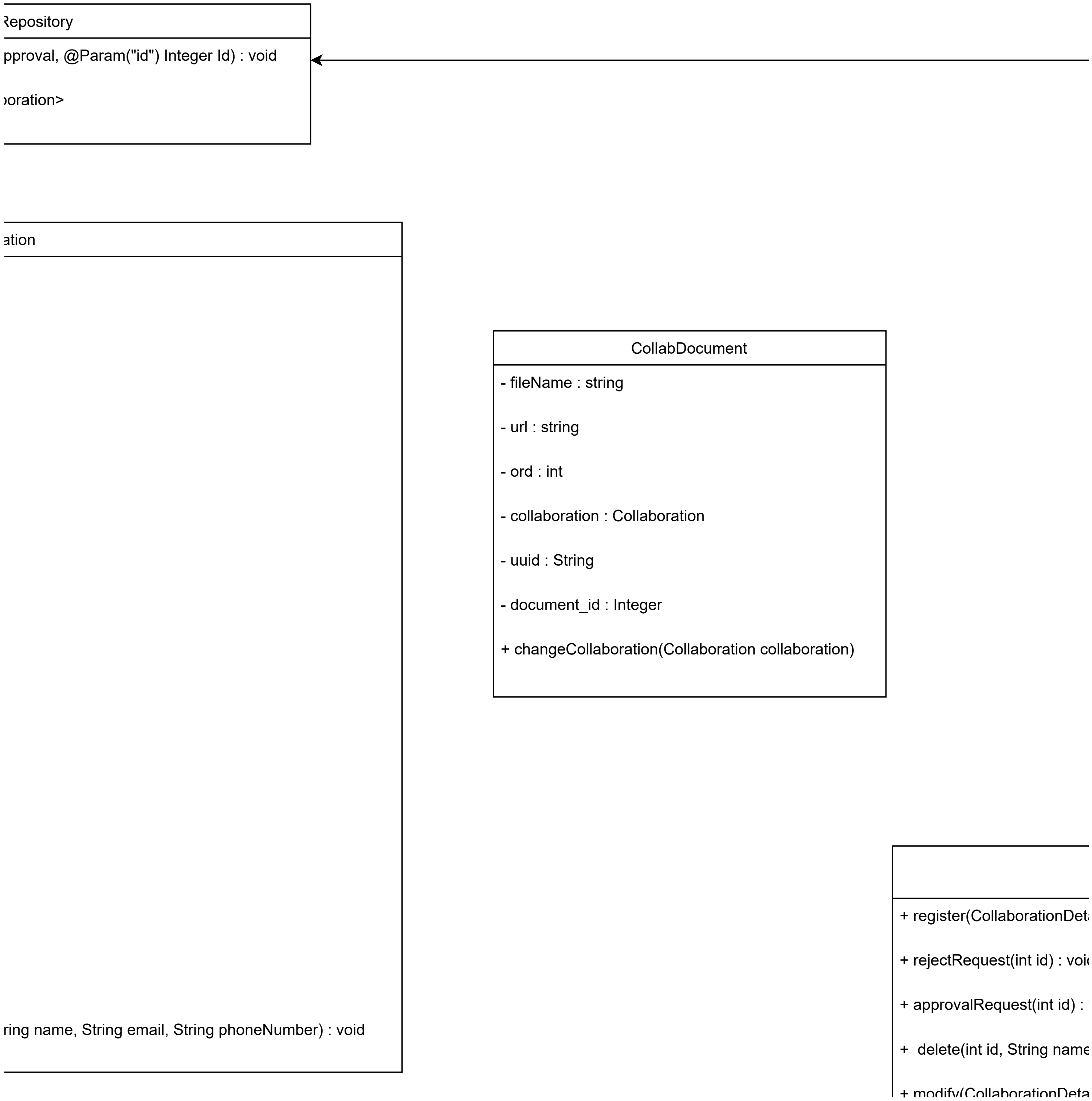
projectOptions : List<ProjectOption>

optionType : String

rewardName : string

changeOption(List<OptionDTO> optionDTOList): void

change(String rewardName,String optionType) : void



Repository

pproval, @Param("id") Integer Id) : void

oration>

ation

ring name, String email, String phoneNumber) : void

CollabDocument

- fileName : string
- url : string
- ord : int
- collaboration : Collaboration
- uuid : String
- document_id : Integer
+ changeCollaboration(Collaboration collaboration)

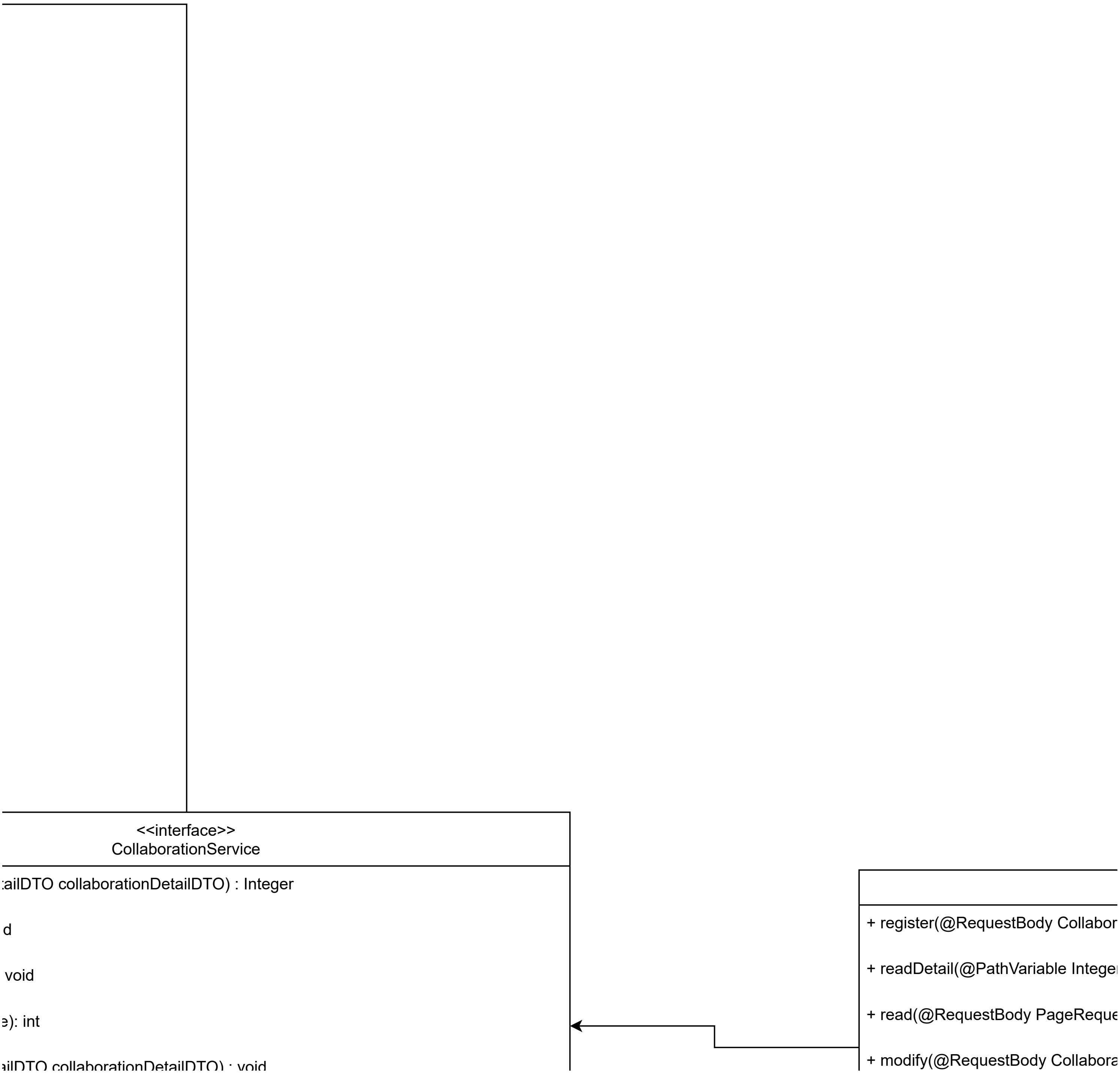
+ register(CollaborationDet

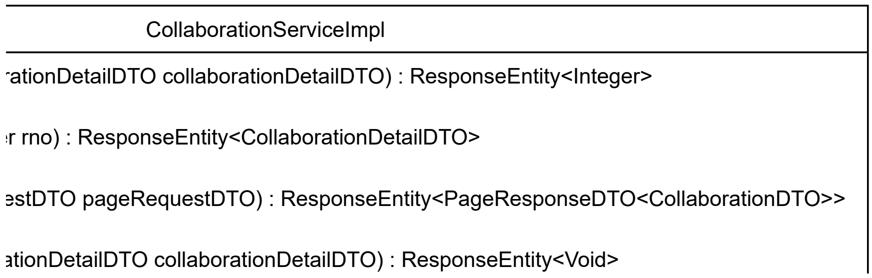
+ rejectRequest(int id) : voi

+ approvalRequest(int id) :

+ delete(int id, String name

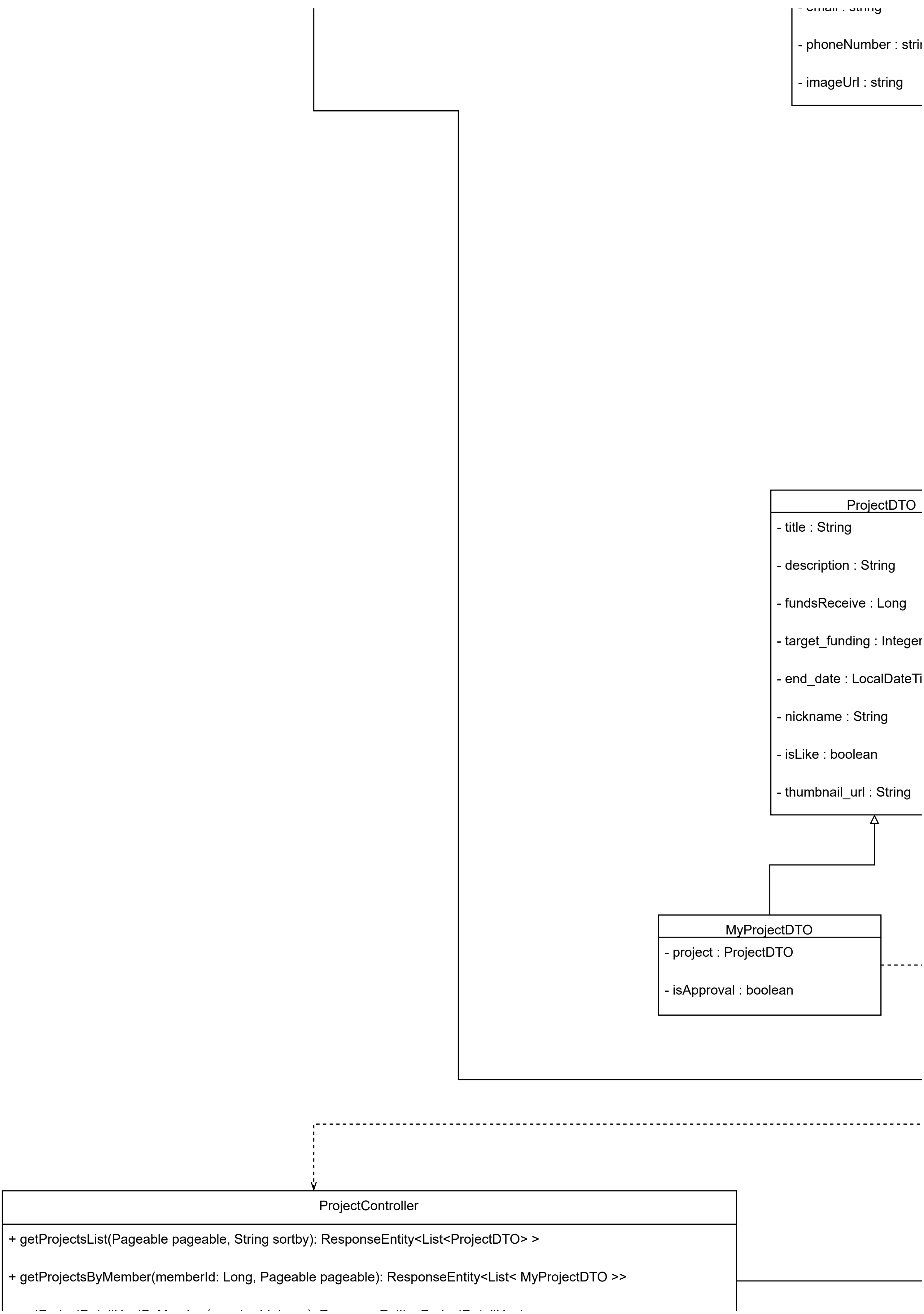
+ modify(CollaborationDeta

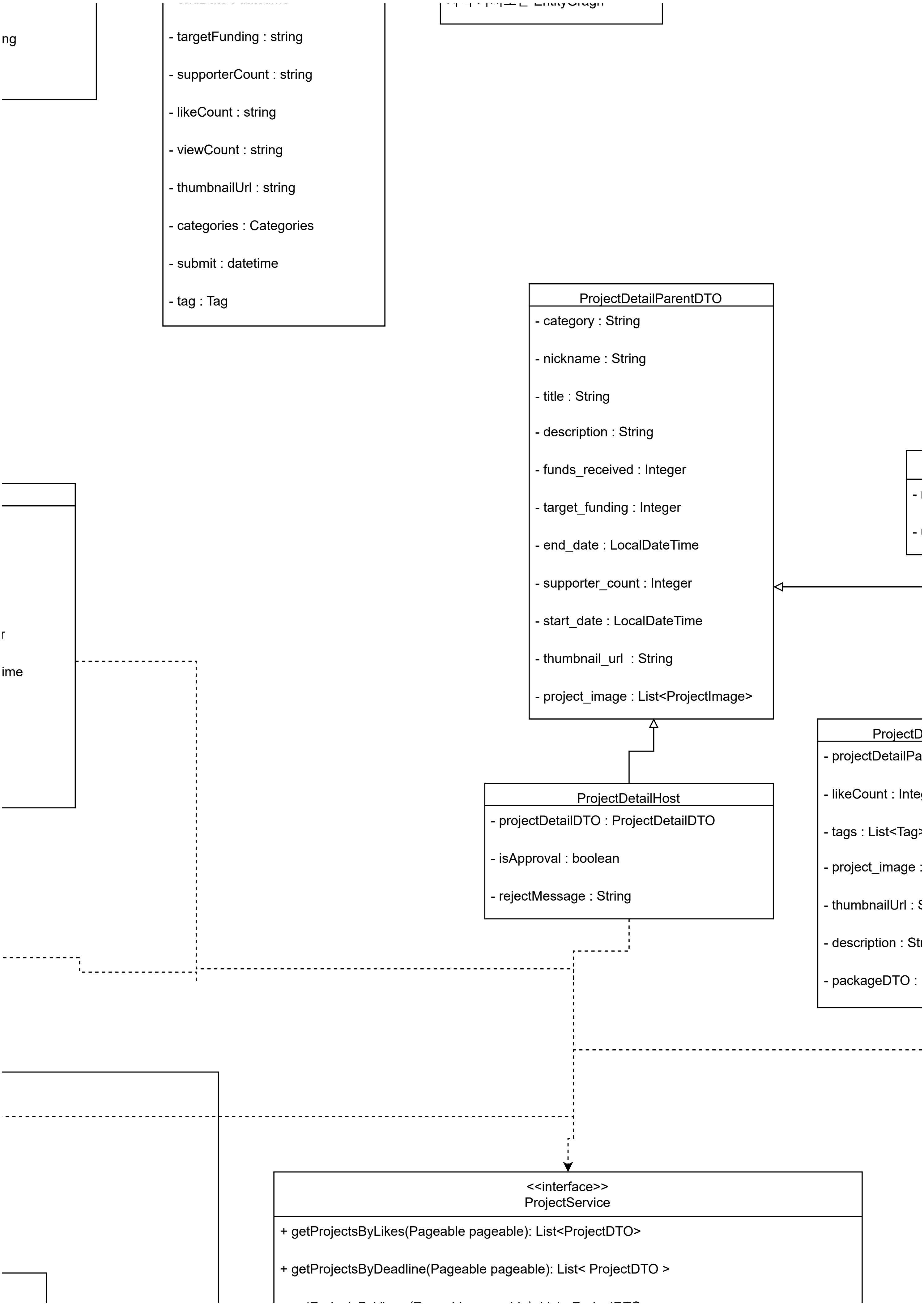

<<interface>>
CollaborationService
:ailDTO collaborationDetailDTO) : Integer
d
void
e): int
ailDTO collaborationDetailDTO) : void
+ register(@RequestBody Collabor
+ readDetail(@PathVariable Intege
+ read(@RequestBody PageReque
+ modify(@RequestBody Collabora

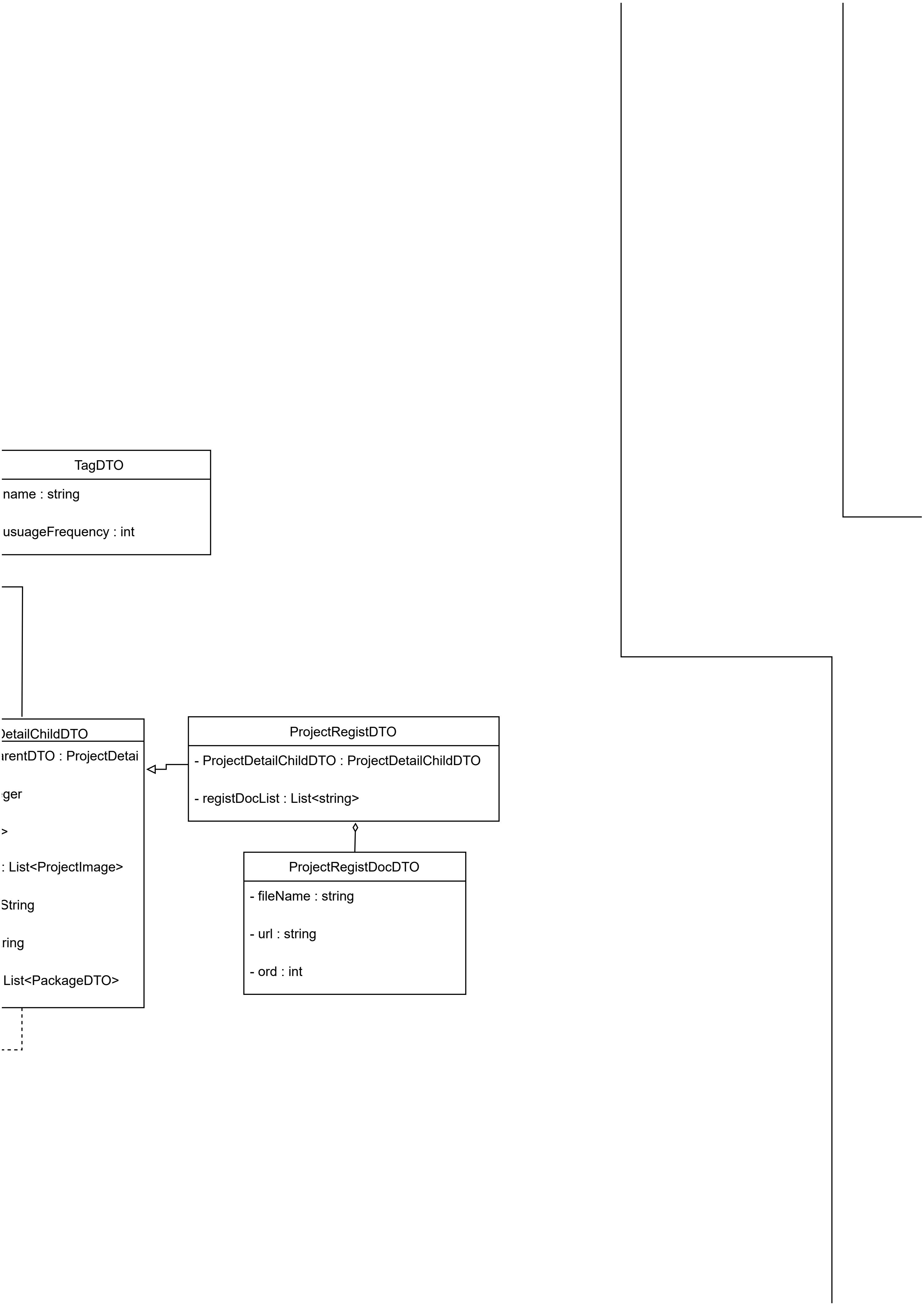


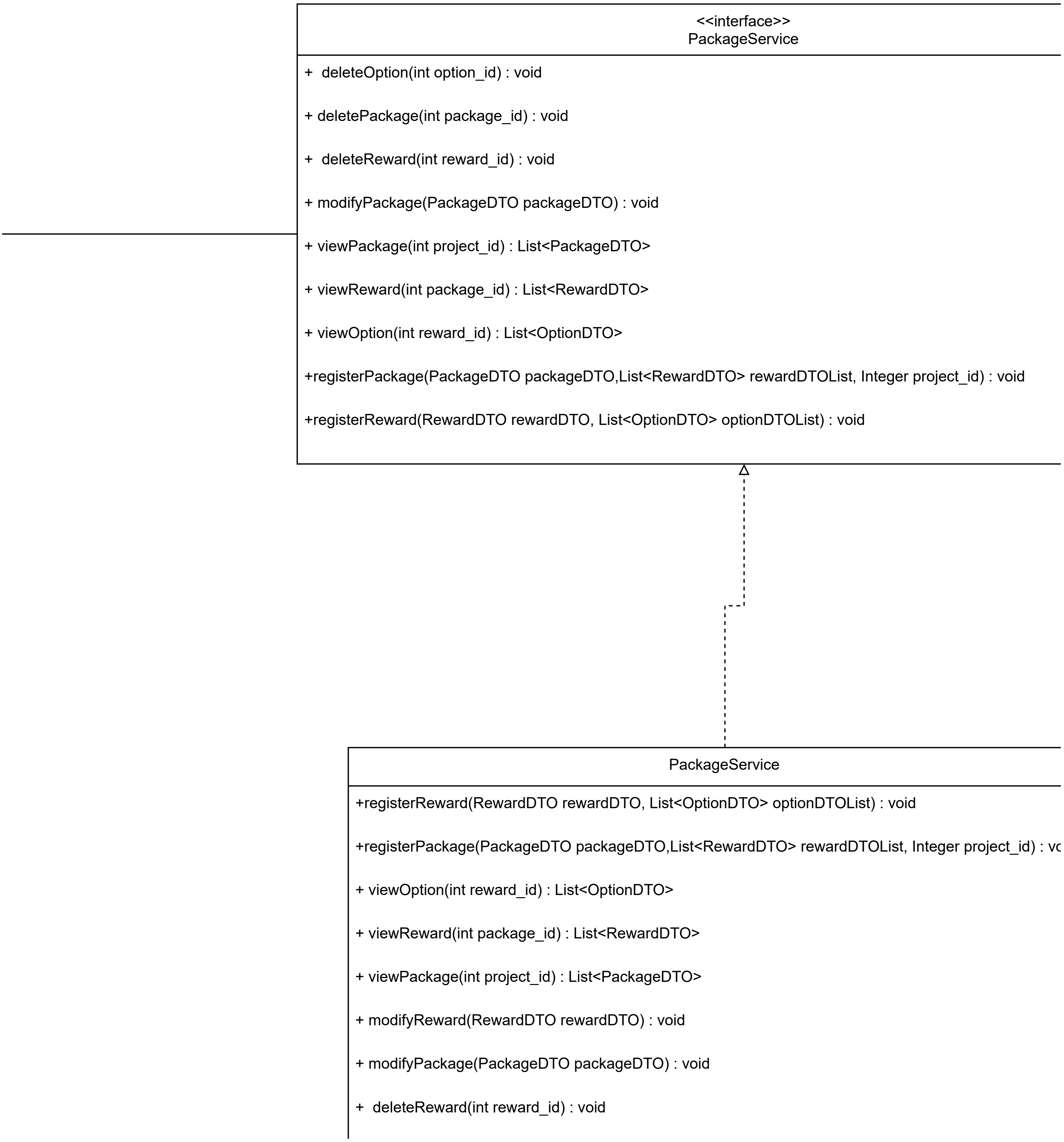
CollaborationServiceImpl

```
rationDetailDTO collaborationDetailDTO) : ResponseEntity<Integer>

r rno) : ResponseEntity<CollaborationDetailDTO>

estDTO pageRequestDTO) : ResponseEntity<PageResponseDTO<CollaborationDTO>>

ationDetailDTO collaborationDetailDTO) : ResponseEntity<Void>
```


- email : string
- phoneNumber : strin
- imageUrl : string
ProjectDTO
- title : String
- description : String
- fundsReceive : Long
- target_funding : Integer
- end_date : LocalDateTi
- nickname : String
- isLike : boolean
- thumbnail_url : String
MyProjectDTO
- project : ProjectDTO
- isApproval : boolean
ProjectController
+ getProjectsList(Pageable pageable, String sortby): ResponseEntity<List<ProjectDTO> >
+ getProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List< MyProjectDTO >>
+ getProjectDetailListByMember(memberId: Long): ResponseEntity<ProjectDetailList


ng
- targetFunding : string
- supporterCount : string
- likeCount : string
- viewCount : string
- thumbnailUrl : string
- categories : Categories
- submit : datetime
- tag : Tag
ProjectDetailParentDTO
- category : String
- nickname : String
- title : String
- description : String
- funds_received : Integer
- target_funding : Integer
- end_date : LocalDateTime
- supporter_count : Integer
- start_date : LocalDateTime
- thumbnail_url : String
- project_image : List<ProjectImage>
ime
ProjectDetailHost
- projectDetailDTO : ProjectDetailDTO
- isApproval : boolean
- rejectMessage : String
ProjectD
- projectDetailPa
- likeCount : Integ
- tags : List<Tag
- project_image :
- thumbnailUrl : S
- description : Str
- packageDTO :
<<interface>>
ProjectService
+ getProjectsByLikes(Pageable pageable): List<ProjectDTO>
+ getProjectsByDeadline(Pageable pageable): List< ProjectDTO >


TagDTO
name : string
usuageFrequency : int
DetailChildDTO
arentDTO : ProjectDetai
ger
>
: List<ProjectImage>
String
ring
List<PackageDTO>
ProjectRegistDTO
- ProjectDetailChildDTO : ProjectDetailChildDTO
- registDocList : List<string>
ProjectRegistDocDTO
- fileName : string
- url : string
- ord : int



<<interface>>
PackageService

+ deleteOption(int option_id) : void

+ deletePackage(int package_id) : void

+ deleteReward(int reward_id) : void

+ modifyPackage(PackageDTO packageDTO) : void

+ viewPackage(int project_id) : List<PackageDTO>

+ viewReward(int package_id) : List<RewardDTO>

+ viewOption(int reward_id) : List<OptionDTO>

+registerPackage(PackageDTO packageDTO,List<RewardDTO> rewardDTOList, Integer project_id) : void

+registerReward(RewardDTO rewardDTO, List<OptionDTO> optionDTOList) : void

PackageService

+registerReward(RewardDTO rewardDTO, List<OptionDTO> optionDTOList) : void

+registerPackage(PackageDTO packageDTO,List<RewardDTO> rewardDTOList, Integer project_id) : vc

+ viewOption(int reward_id) : List<OptionDTO>

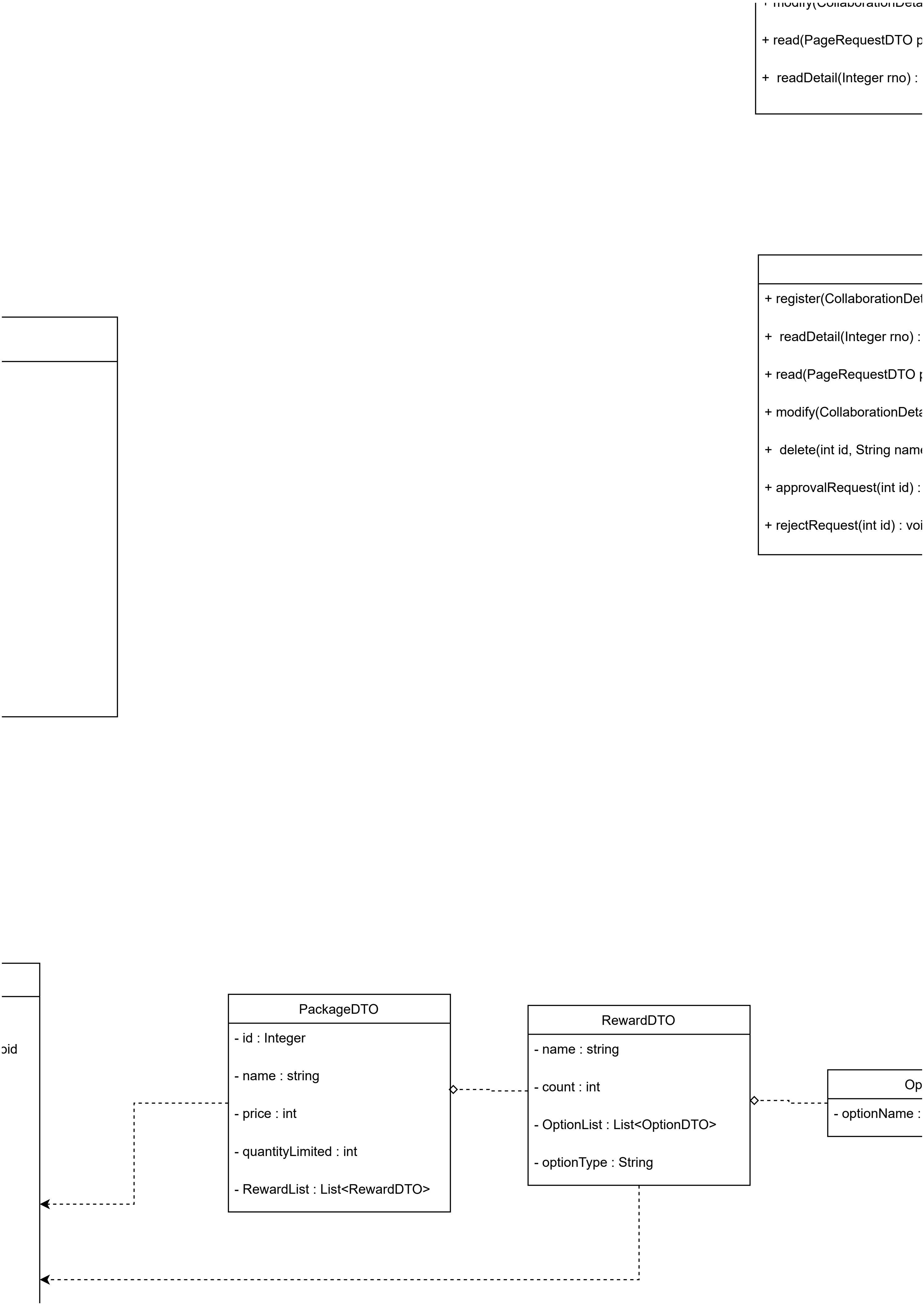
+ viewReward(int package_id) : List<RewardDTO>

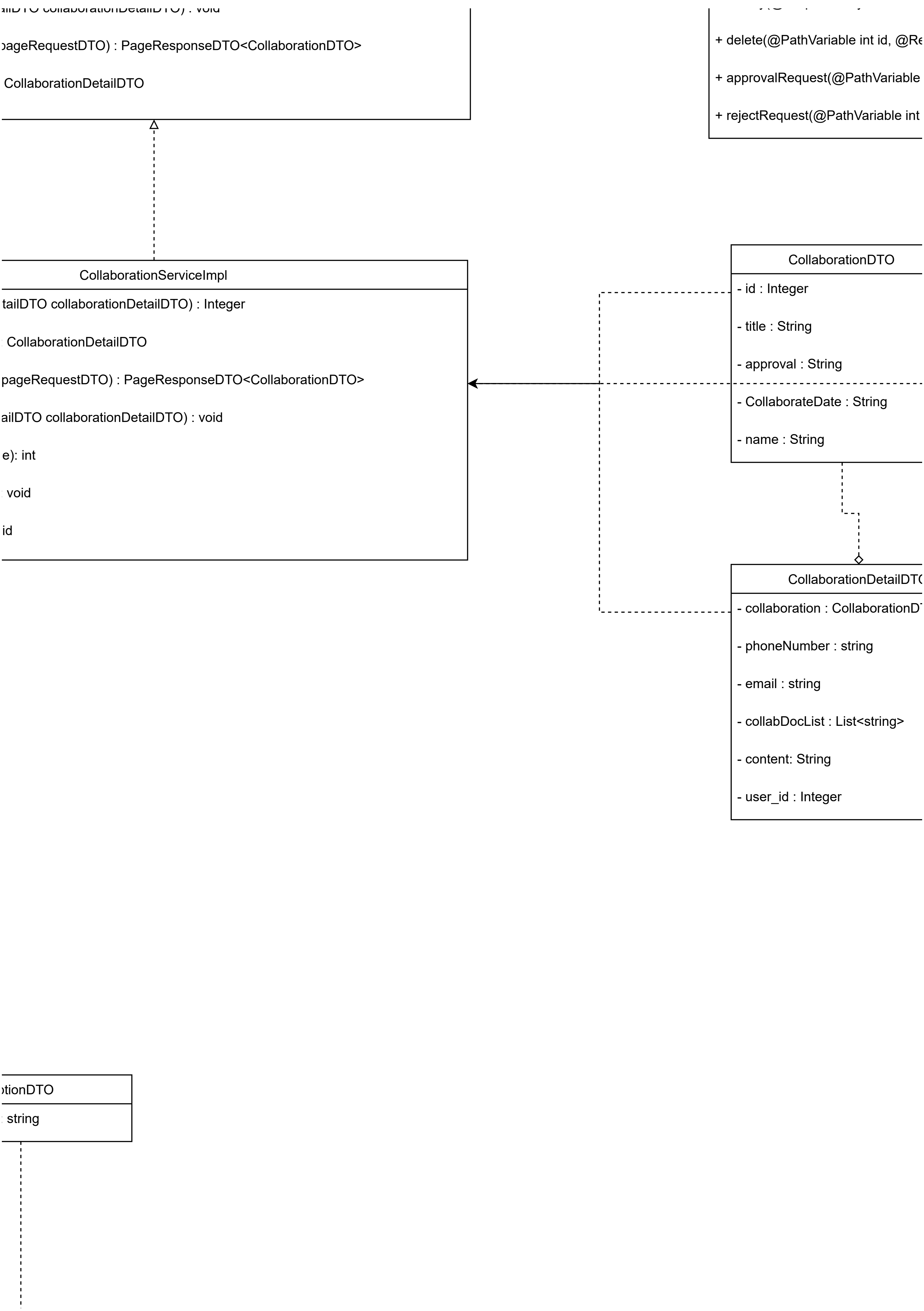
+ viewPackage(int project_id) : List<PackageDTO>

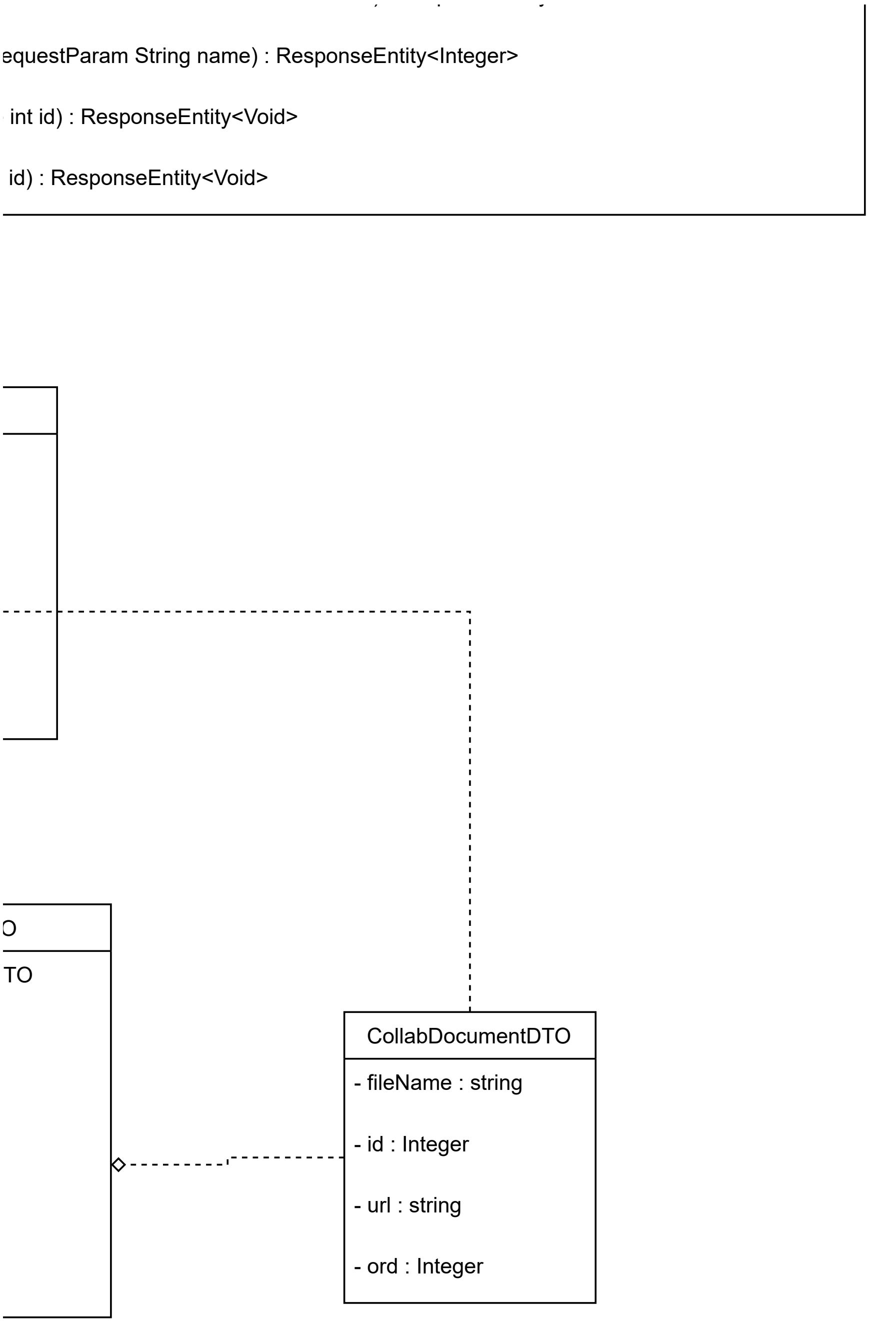
+ modifyReward(RewardDTO rewardDTO) : void

+ modifyPackage(PackageDTO packageDTO) : void

+ deleteReward(int reward_id) : void

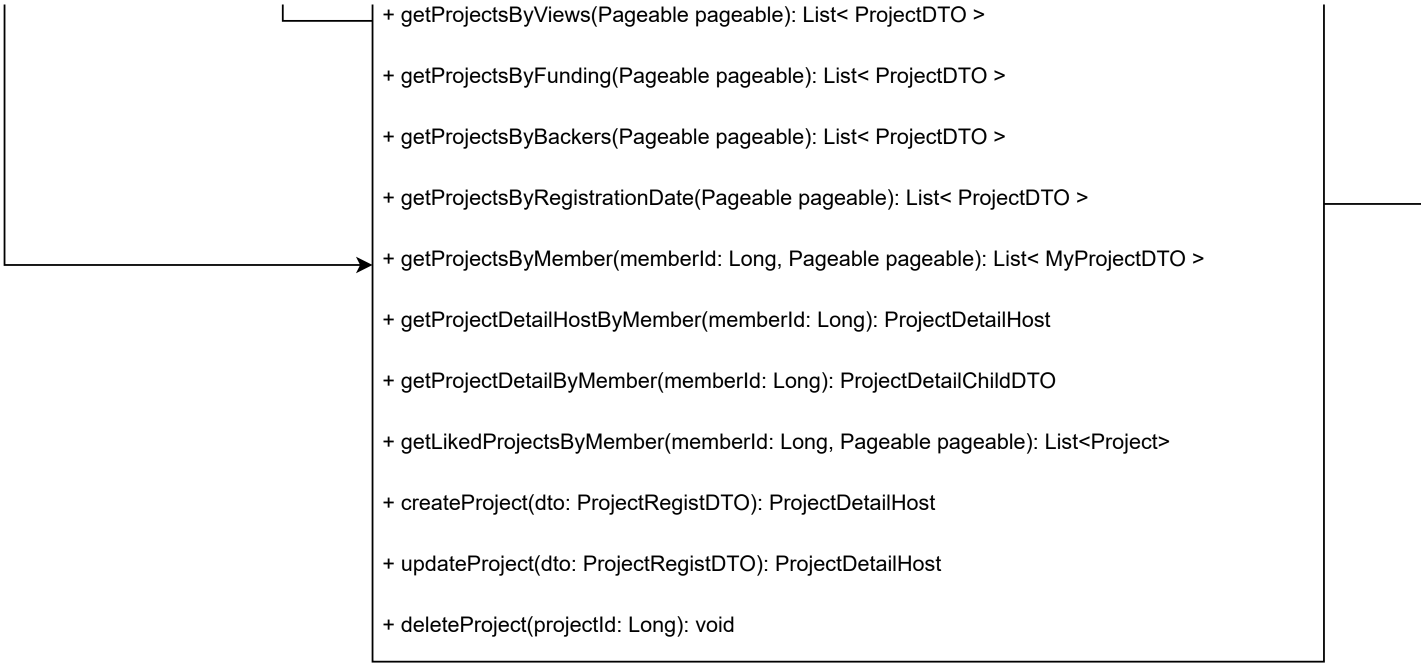

+ modify(CollaborationDeta
+ read(PageRequestDTO p
+ readDetail(Integer rno) :
+ register(CollaborationDet
+ readDetail(Integer rno) :
+ read(PageRequestDTO p
+ modify(CollaborationDeta
+ delete(int id, String name
+ approvalRequest(int id) :
+ rejectRequest(int id) : voi
oid
PackageDTO
- id : Integer
- name : string
- price : int
- quantityLimited : int
- RewardList : List<RewardDTO>
RewardDTO
- name : string
- count : int
- OptionList : List<OptionDTO>
- optionType : String
Op
- optionName :


ailDTO collaborationDetailDTO) : void
ɔageRequestDTO) : PageResponseDTO<CollaborationDTO>
CollaborationDetailDTO
+ delete(@PathVariable int id, @Re
+ approvalRequest(@PathVariable
+ rejectRequest(@PathVariable int
CollaborationServiceImpl
tailDTO collaborationDetailDTO) : Integer
: CollaborationDetailDTO
pageRequestDTO) : PageResponseDTO<CollaborationDTO>
ailDTO collaborationDetailDTO) : void
e): int
: void
id
CollaborationDTO
- id : Integer
- title : String
- approval : String
- CollaborateDate : String
- name : String
CollaborationDetailDT(
- collaboration : CollaborationD
- phoneNumber : string
- email : string
- collabDocList : List<string>
- content: String
- user_id : Integer
ɔtionDTO
: string


equestParam String name) : ResponseEntity<Integer>
int id) : ResponseEntity<Void>
id) : ResponseEntity<Void>
O
TO
CollabDocumentDTO
- fileName : string
- id : Integer
- url : string
- ord : Integer

```
+ getProjectDetailHostByMember(memberId: Long): ResponseEntity<ProjectDetailHost>

+ getProjectDetailByMember(memberId: Long): ResponseEntity<ProjectDetailChildDTO>

+ getLikedProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>>

+ createProject(dto: ProjectRegistDTO): ResponseEntity<ProjectDetailHost>

+ updateProject(dto: ProjectRegistDTO): ResponseEntity<ProjectDetailHost>

+ deleteProject(projectId: Long): void
```

```
+---------------------------------------------------------------+
| + getProjectsByLikes(Pageable pageable): ResponseEntity<List<Project>> | <-- 1. 관심 순 조회
| + getProjectsByDeadline(Pageable pageable): ResponseEntity<List<Project>> | <-- 2. 마감일 순 조회
| + getProjectsByViews(Pageable pageable): ResponseEntity<List<Project>> | <-- 3. 조회수 순 조회
| + getProjectsByFunding(Pageable pageable): ResponseEntity<List<Project>> | <-- 4. 후원금 순 조회
| + getProjectsByBackers(Pageable pageable): ResponseEntity<List<Project>> | <-- 5. 후원자 순 조회
| + getProjectsByRegistrationDate(Pageable pageable): ResponseEntity<List<Project>> | <-- 6. 등록일 순 조회
| + getProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>> | <-- 7. 특정 회원의 프로젝트 조회
| + getProjectDetailByMember(memberId: Long): ResponseEntity<ProjectDetailDTO> | <-- 8. 특정 회원의 프로젝트 상세 보기
| + getLikedProjectsByMember(memberId: Long, Pageable pageable): ResponseEntity<List<Project>> | <-- 9. 특정 회원의 관심 프로젝트 조회
| + createProject(dto: ProjectDTO): ResponseEntity<Project> | <-- 10. 프로젝트 생성
| + updateProject(dto: ProjectDTO): ResponseEntity<Project> | <-- 11. 프로젝트 수정
| + deleteProject(projectId: Long): ResponseEntity<Void> | <-- 12. 프로젝트 삭제
+---------------------------------------------------------------+
```



+ getProjectsByViews(Pageable pageable): List< ProjectDTO >

+ getProjectsByFunding(Pageable pageable): List< ProjectDTO >

+ getProjectsByBackers(Pageable pageable): List< ProjectDTO >

+ getProjectsByRegistrationDate(Pageable pageable): List< ProjectDTO >

+ getProjectsByMember(memberId: Long, Pageable pageable): List< MyProjectDTO >

+ getProjectDetailHostByMember(memberId: Long): ProjectDetailHost

+ getProjectDetailByMember(memberId: Long): ProjectDetailChildDTO

+ getLikedProjectsByMember(memberId: Long, Pageable pageable): List<Project>

+ createProject(dto: ProjectRegistDTO): ProjectDetailHost

+ updateProject(dto: ProjectRegistDTO): ProjectDetailHost

+ deleteProject(projectId: Long): void

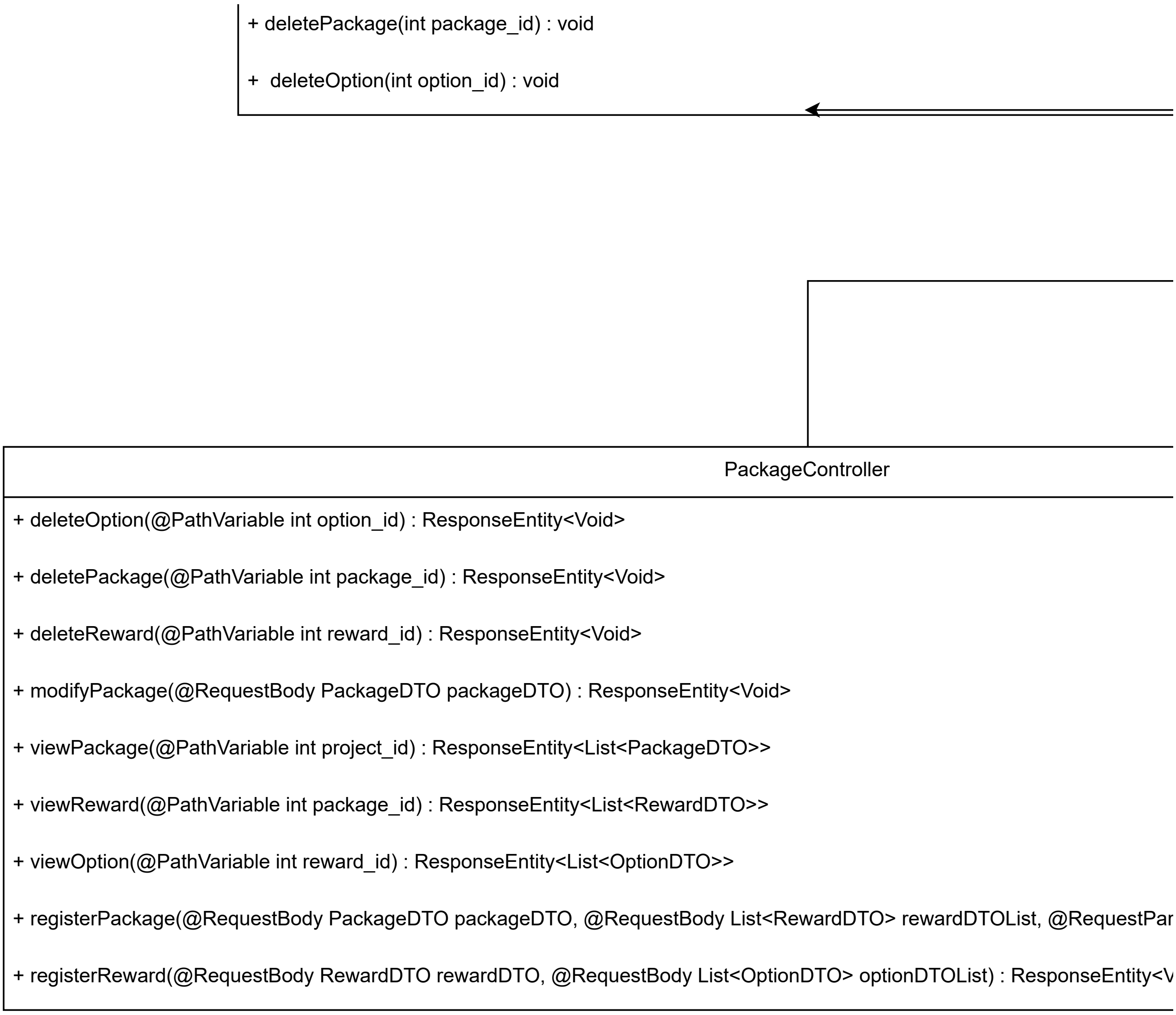
이건 그냥 메모... 뭔지 모를까봐요...ㅎㅎ

```
| + getProjectsByLikes(Pageable pageable): List<Project>     |  <-- 1. 관심표시가 많은 순으로 프로젝트 조회 (페이지네이
| + getProjectsByDeadline(Pageable pageable): List<Project>  |  <-- 2. 마감일에 가까운 순으로 프로젝트 조회 (페이지네
| + getProjectsByViews(Pageable pageable): List<Project>     |  <-- 3. 조회수가 많은 순으로 프로젝트 조회 (페이지네이션
| + getProjectsByFunding(Pageable pageable): List<Project>   |  <-- 4. 후원금액이 많은 순으로 프로젝트 조회 (페이지네
| + getProjectsByBackers(Pageable pageable): List<Project>   |  <-- 5. 후원자 수가 많은 순으로 프로젝트 조회 (페이지네
| + getProjectsByRegistrationDate(Pageable pageable): List<Project> |  <-- 6. 등록일 순으로 프로젝트 조회 (페이지네이
| + getProjectsByMember(memberId: Long, Pageable pageable): List<Project> |  <-- 7. 특정 회원의 프로젝트 조회 (memb
| + getProjectDetailByMember(memberId: Long): ProjectDetailDTO |  <-- 8. 특정 회원의 프로젝트 상세 보기 (memberId)
| + getLikedProjectsByMember(memberId: Long, Pageable pageable): List<Project> |  <-- 9. 특정 회원의 관심 프로젝트
| + createProject(dto: ProjectDTO): Project                  |  <-- 10. 프로젝트 생성 (dto)
| + updateProject(dto: ProjectDTO): Project                  |  <-- 11. 프로젝트 수정 (dto)
| + deleteProject(projectId: Long): void                     |  <-- 12. 프로젝트 삭제 (projectId
```

션)
이션)
)
이션)
이션)
션)
erId)

조회 (memberId)

따로 가져가지는 않고 여기서 작업할게용!



+ deletePackage(int package_id) : void

+ deleteOption(int option_id) : void

PackageController

+ deleteOption(@PathVariable int option_id) : ResponseEntity<Void>

+ deletePackage(@PathVariable int package_id) : ResponseEntity<Void>

+ deleteReward(@PathVariable int reward_id) : ResponseEntity<Void>

+ modifyPackage(@RequestBody PackageDTO packageDTO) : ResponseEntity<Void>

+ viewPackage(@PathVariable int project_id) : ResponseEntity<List<PackageDTO>>

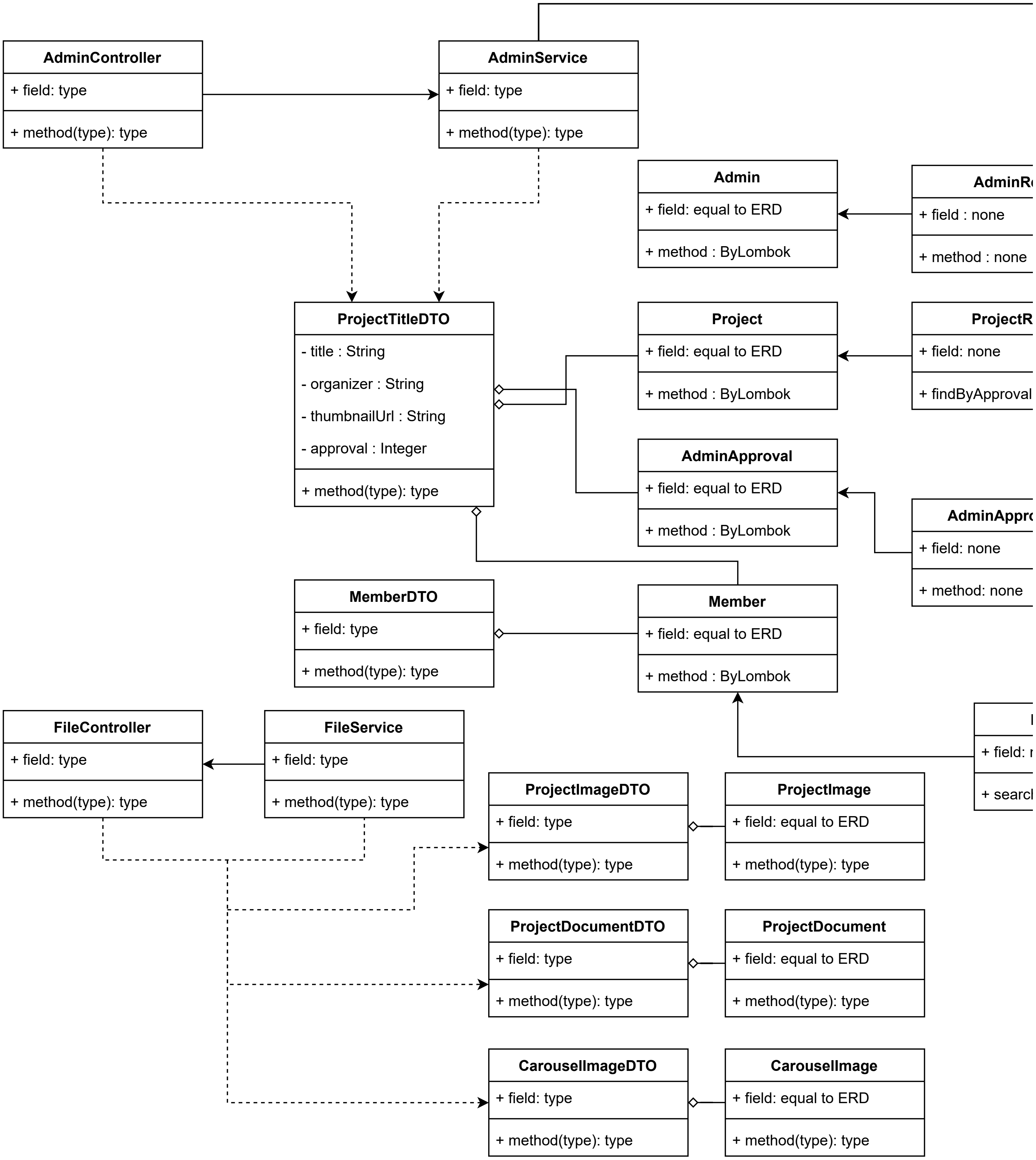
+ viewReward(@PathVariable int package_id) : ResponseEntity<List<RewardDTO>>

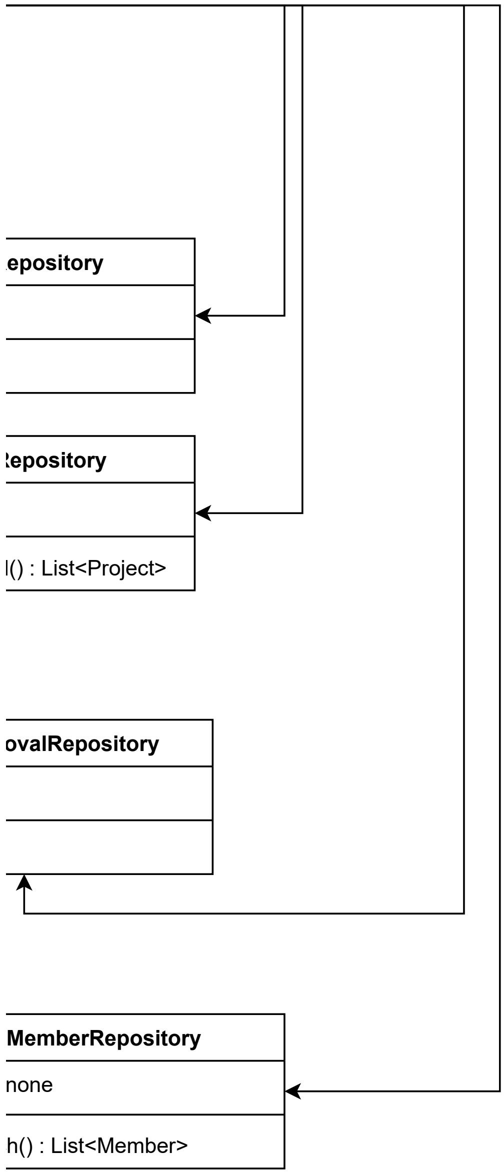
+ viewOption(@PathVariable int reward_id) : ResponseEntity<List<OptionDTO>>

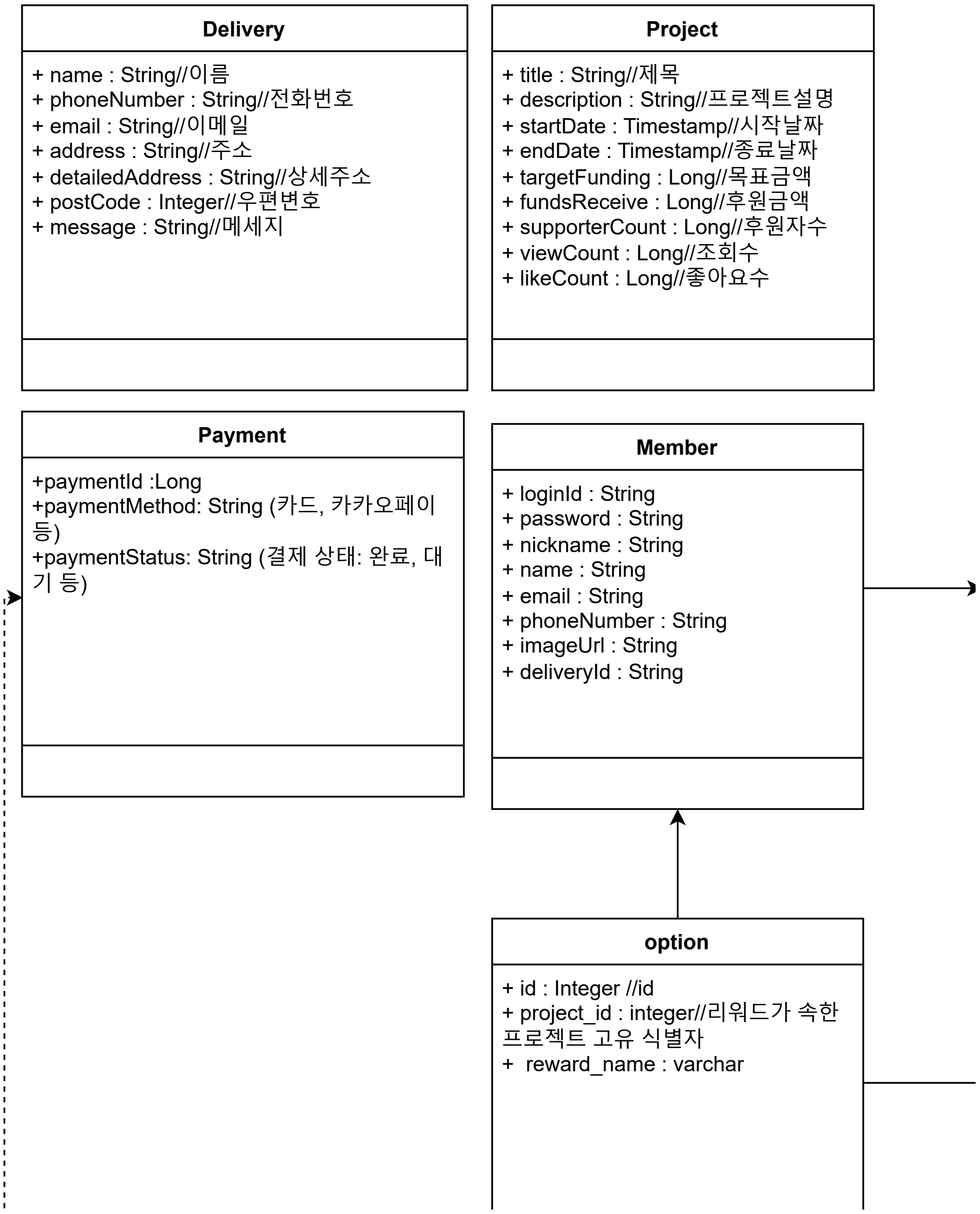
+ registerPackage(@RequestBody PackageDTO packageDTO, @RequestBody List<RewardDTO> rewardDTOList, @RequestPar

+ registerReward(@RequestBody RewardDTO rewardDTO, @RequestBody List<OptionDTO> optionDTOList) : ResponseEntity<V

ram Integer project_id) : ResponseEntity<Void>
/oid>


AdminController
+ field: type
+ method(type): type
AdminService
+ field: type
+ method(type): type
Admin
+ field: equal to ERD
+ method : ByLombok
AdminR
+ field : none
+ method : none
ProjectTitleDTO
- title : String
- organizer : String
- thumbnailUrl : String
- approval : Integer
+ method(type): type
Project
+ field: equal to ERD
+ method : ByLombok
ProjectR
+ field: none
+ findByApproval
AdminApproval
+ field: equal to ERD
+ method : ByLombok
AdminAppro
+ field: none
+ method: none
MemberDTO
+ field: type
+ method(type): type
Member
+ field: equal to ERD
+ method : ByLombok
I
+ field: r
+ searcl
FileController
+ field: type
+ method(type): type
FileService
+ field: type
+ method(type): type
ProjectImageDTO
+ field: type
+ method(type): type
ProjectImage
+ field: equal to ERD
+ method(type): type
ProjectDocumentDTO
+ field: type
+ method(type): type
ProjectDocument
+ field: equal to ERD
+ method(type): type
CarouselImageDTO
+ field: type
+ method(type): type
CarouselImage
+ field: equal to ERD
+ method(type): type


epository
Repository
l() : List<Project>
ovalRepository
MemberRepository
none
h() : List<Member>



Delivery

+ name : String//이름
+ phoneNumber : String//전화번호
+ email : String//이메일
+ address : String//주소
+ detailedAddress : String//상세주소
+ postCode : Integer//우편번호
+ message : String//메세지

Project

+ title : String//제목
+ description : String//프로젝트설명
+ startDate : Timestamp//시작날짜
+ endDate : Timestamp//종료날짜
+ targetFunding : Long//목표금액
+ fundsReceive : Long//후원금액
+ supporterCount : Long//후원자수
+ viewCount : Long//조회수
+ likeCount : Long//좋아요수

Payment

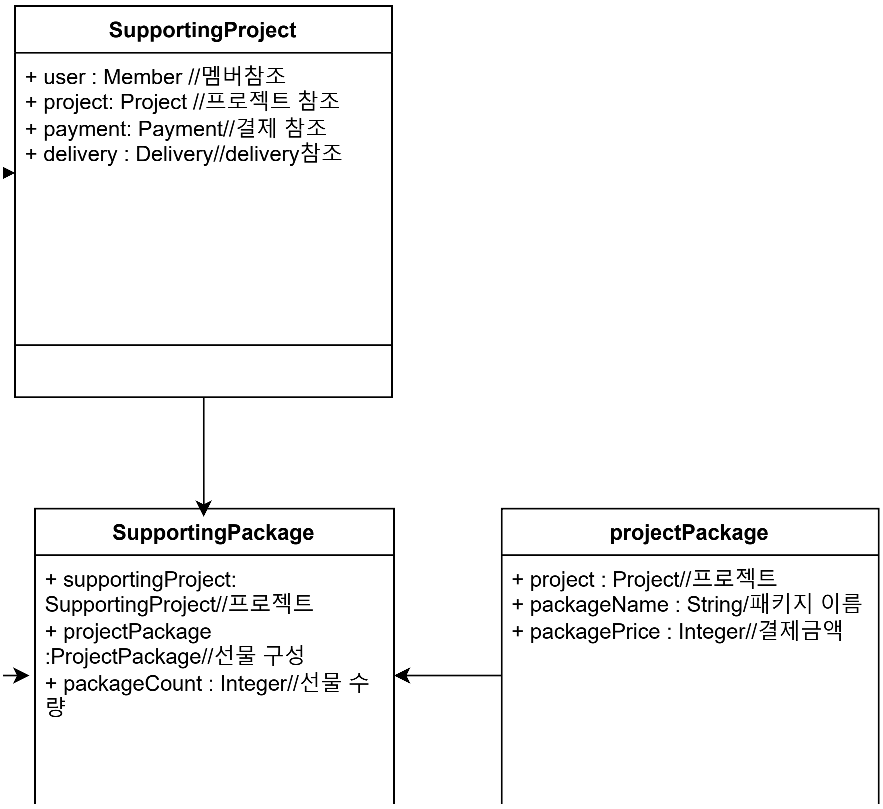
+paymentId :Long
+paymentMethod: String (카드, 카카오페이 등)
+paymentStatus: String (결제 상태: 완료, 대기 등)

Member

+ loginId : String
+ password : String
+ nickname : String
+ name : String
+ email : String
+ phoneNumber : String
+ imageUrl : String
+ deliveryId : String

option

+ id : Integer //id
+ project_id : integer//리워드가 속한 프로젝트 고유 식별자
+ reward_name : varchar



SupportingProject

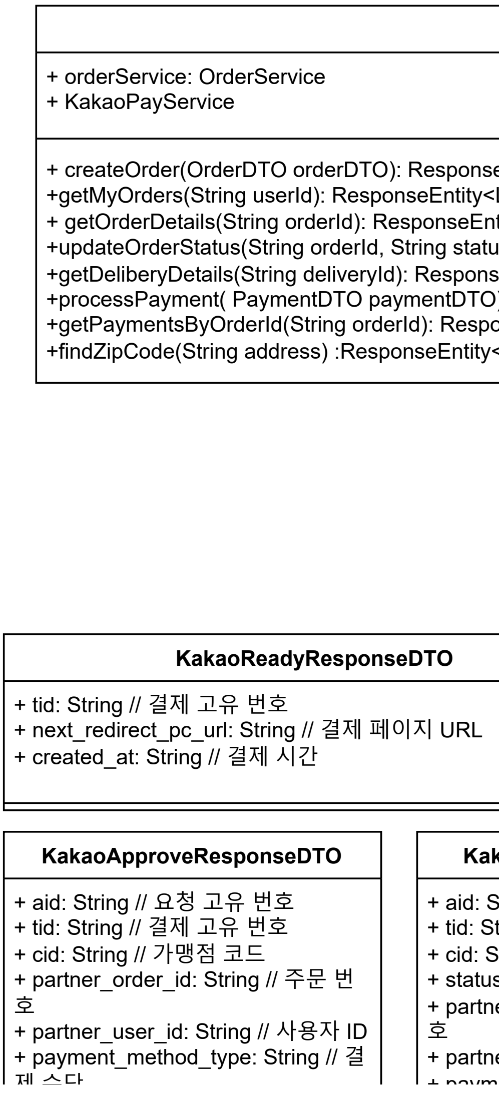
+ user : Member //멤버참조
+ project: Project //프로젝트 참조
+ payment: Payment//결제 참조
+ delivery : Delivery//delivery참조

SupportingPackage

+ supportingProject: SupportingProject//프로젝트
+ projectPackage :ProjectPackage//선물 구성
+ packageCount : Integer//선물 수량

projectPackage

+ project : Project//프로젝트
+ packageName : String/패키지 이름
+ packagePrice : Integer//결제금액



+ orderService: OrderService
+ KakaoPayService

+ createOrder(OrderDTO orderDTO): Response
+getMyOrders(String userId): ResponseEntity<I
+ getOrderDetails(String orderId): ResponseEnt
+updateOrderStatus(String orderId, String statu
+getDeliberyDetails(String deliveryId): Respons
+processPayment(PaymentDTO paymentDTO
+getPaymentsByOrderId(String orderId): Respo
+findZipCode(String address) :ResponseEntity<

KakaoReadyResponseDTO

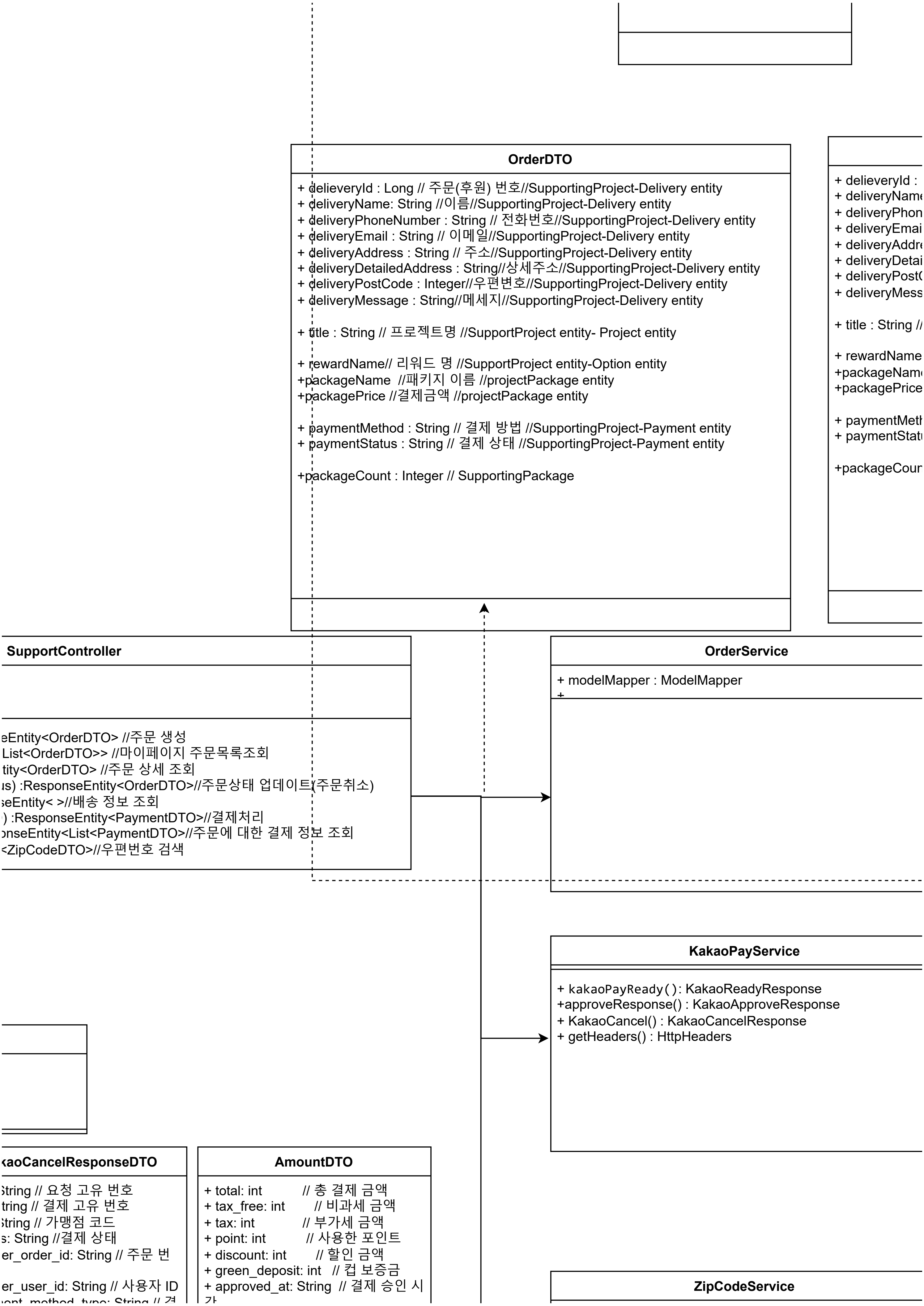
+ tid: String // 결제 고유 번호
+ next_redirect_pc_url: String // 결제 페이지 URL
+ created_at: String // 결제 시간

KakaoApproveResponseDTO

+ aid: String // 요청 고유 번호
+ tid: String // 결제 고유 번호
+ cid: String // 가맹점 코드
+ partner_order_id: String // 주문 번호
+ partner_user_id: String // 사용자 ID
+ payment_method_type: String // 결제 수단

Kak

+ aid: S
+ tid: St
+ cid: S
+ status
+ partne
호
+ partne
+ paym



OrderDTO

+ delieveryId : Long // 주문(후원) 번호//SupportingProject-Delivery entity
+ deliveryName: String //이름//SupportingProject-Delivery entity
+ deliveryPhoneNumber : String // 전화번호//SupportingProject-Delivery entity
+ deliveryEmail : String // 이메일//SupportingProject-Delivery entity
+ deliveryAddress : String // 주소//SupportingProject-Delivery entity
+ deliveryDetailedAddress : String//상세주소//SupportingProject-Delivery entity
+ deliveryPostCode : Integer//우편번호//SupportingProject-Delivery entity
+ deliveryMessage : String//메세지//SupportingProject-Delivery entity

+ title : String // 프로젝트명 //SupportProject entity- Project entity

+ rewardName// 리워드 명 //SupportProject entity-Option entity
+packageName //패키지 이름 //projectPackage entity
+packagePrice //결제금액 //projectPackage entity

+ paymentMethod : String // 결제 방법 //SupportingProject-Payment entity
+ paymentStatus : String // 결제 상태 //SupportingProject-Payment entity

+packageCount : Integer // SupportingPackage

+ delieveryId :
+ deliveryNam
+ deliveryPhon
+ deliveryEmai
+ deliveryAddr
+ deliveryDetai
+ deliveryPost
+ deliveryMess

+ title : String /

+ rewardName
+packageNam
+packagePrice

+ paymentMeth
+ paymentStatu

+packageCour

SupportController

eEntity<OrderDTO> //주문 생성
List<OrderDTO>> //마이페이지 주문목록조회
tity<OrderDTO> //주문 상세 조회
ıs) :ResponseEntity<OrderDTO>//주문상태 업데이트(주문취소)
seEntity< >//배송 정보 조회
) :ResponseEntity<PaymentDTO>//결제처리
onseEntity<List<PaymentDTO>//주문에 대한 결제 정보 조회
<ZipCodeDTO>//우편번호 검색

OrderService

+ modelMapper : ModelMapper
+

KakaoPayService

+ kakaoPayReady(): KakaoReadyResponse
+approveResponse() : KakaoApproveResponse
+ KakaoCancel() : KakaoCancelResponse
+ getHeaders() : HttpHeaders

‹aoCancelResponseDTO

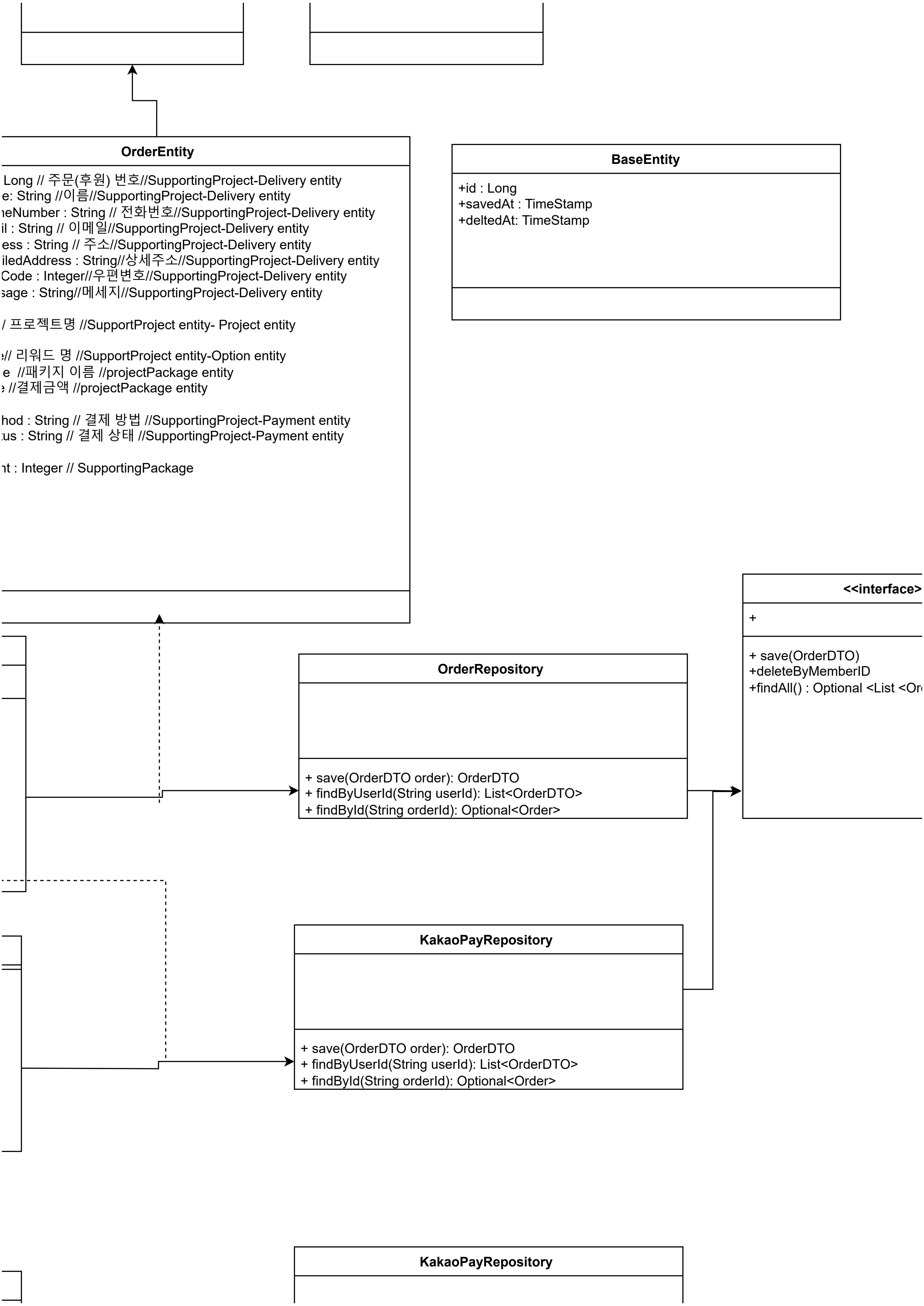
String // 요청 고유 번호
tring // 결제 고유 번호
String // 가맹점 코드
s: String //결제 상태
er_order_id: String // 주문 번

er_user_id: String // 사용자 ID

AmountDTO

+ total: int // 총 결제 금액
+ tax_free: int // 비과세 금액
+ tax: int // 부가세 금액
+ point: int // 사용한 포인트
+ discount: int // 할인 금액
+ green_deposit: int // 컵 보증금
+ approved_at: String // 결제 승인 시간

ZipCodeService



OrderEntity

Long // 주문(후원) 번호//SupportingProject-Delivery entity
e: String //이름//SupportingProject-Delivery entity
ıeNumber : String // 전화번호//SupportingProject-Delivery entity
il : String // 이메일//SupportingProject-Delivery entity
ess : String // 주소//SupportingProject-Delivery entity
iledAddress : String//상세주소//SupportingProject-Delivery entity
Code : Integer//우편번호//SupportingProject-Delivery entity
sage : String//메세지//SupportingProject-Delivery entity

/ 프로젝트명 //SupportProject entity- Project entity

e// 리워드 명 //SupportProject entity-Option entity
e //패키지 이름 //projectPackage entity
e //결제금액 //projectPackage entity

hod : String // 결제 방법 //SupportingProject-Payment entity
tus : String // 결제 상태 //SupportingProject-Payment entity

ıt : Integer // SupportingPackage

BaseEntity

+id : Long
+savedAt : TimeStamp
+deltedAt: TimeStamp

<<interface>

+

+ save(OrderDTO)
+deleteByMemberID
+findAll() : Optional <List <Or

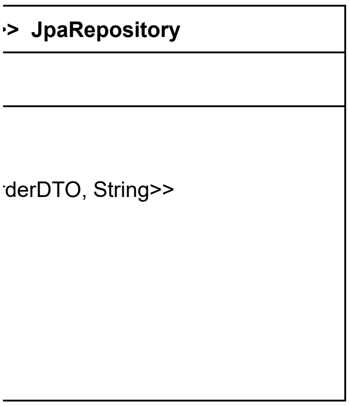
OrderRepository

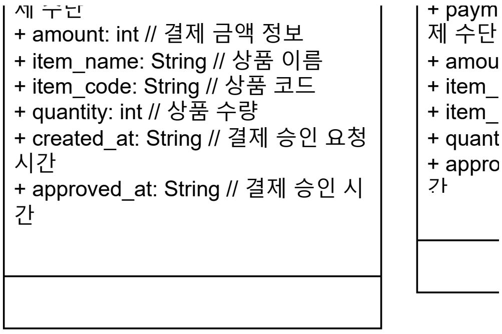
+ save(OrderDTO order): OrderDTO
+ findByUserId(String userId): List<OrderDTO>
+ findById(String orderId): Optional<Order>

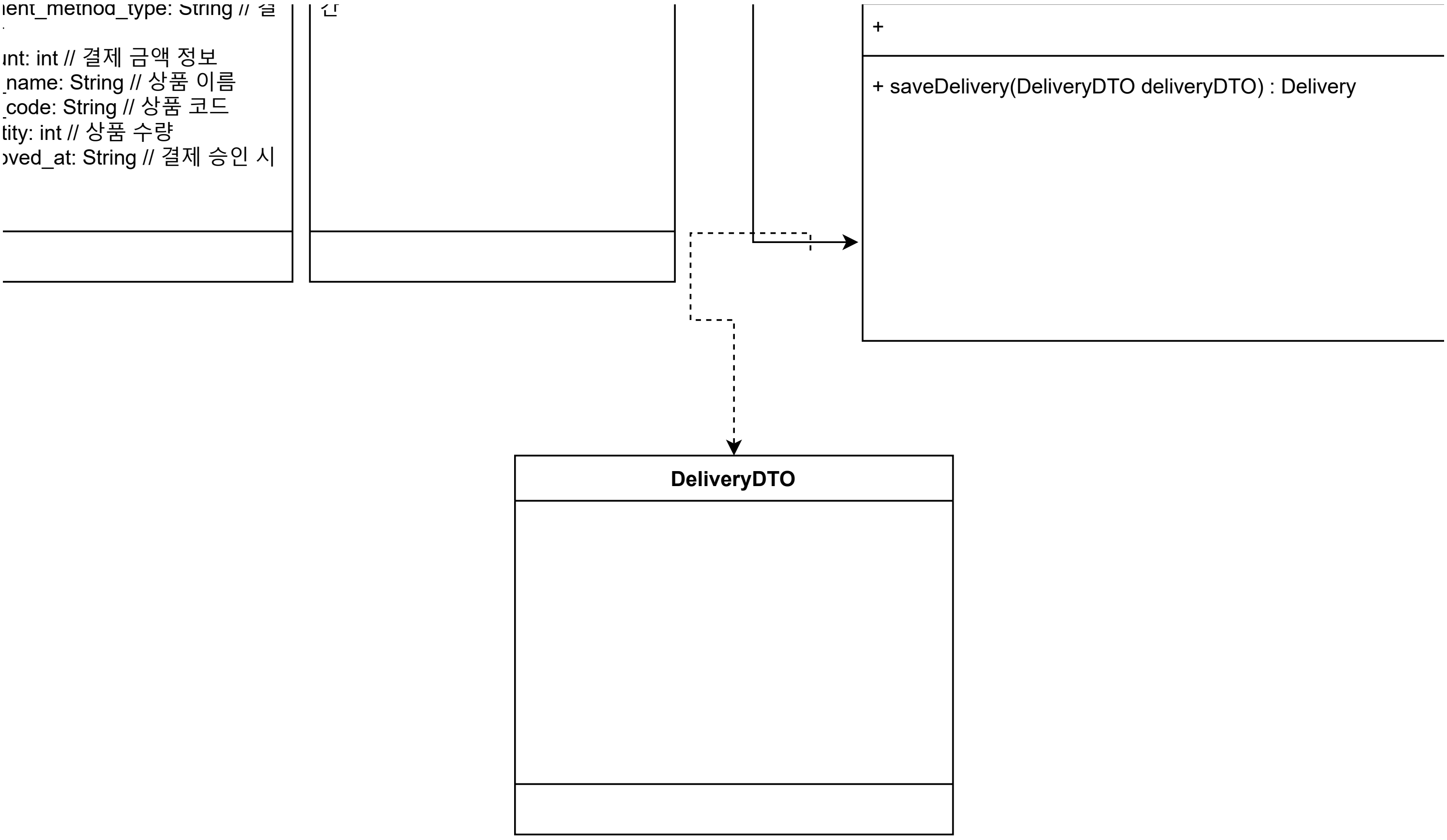
KakaoPayRepository

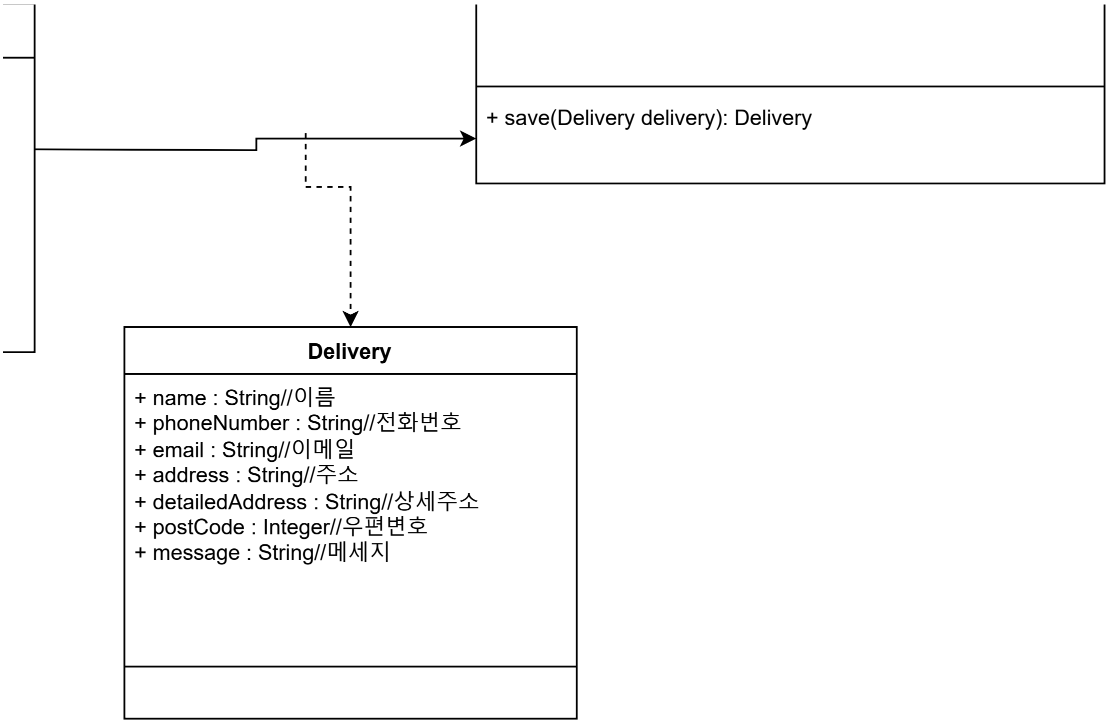
+ save(OrderDTO order): OrderDTO
+ findByUserId(String userId): List<OrderDTO>
+ findById(String orderId): Optional<Order>

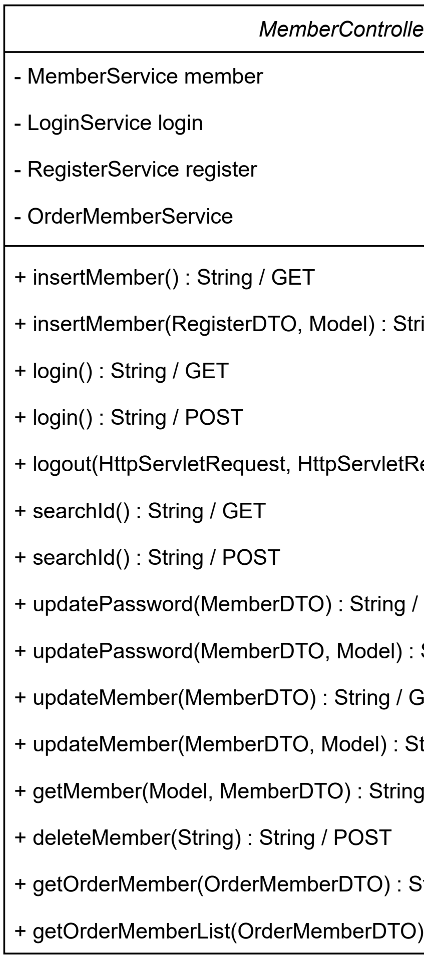
KakaoPayRepository


> JpaRepository
rderDTO, String>>


제 수단
+ amount: int // 결제 금액 정보
+ item_name: String // 상품 이름
+ item_code: String // 상품 코드
+ quantity: int // 상품 수량
+ created_at: String // 결제 승인 요청 시간
+ approved_at: String // 결제 승인 시간
+ paym
제 수단
+ amou
+ item_
+ item_
+ quant
+ appro
간


ıent_method_type: String // 결
ınt: int // 결제 금액 정보
_name: String // 상품 이름
_code: String // 상품 코드
tity: int // 상품 수량
›ved_at: String // 결제 승인 시
간
+
+ saveDelivery(DeliveryDTO deliveryDTO) : Delivery
DeliveryDTO

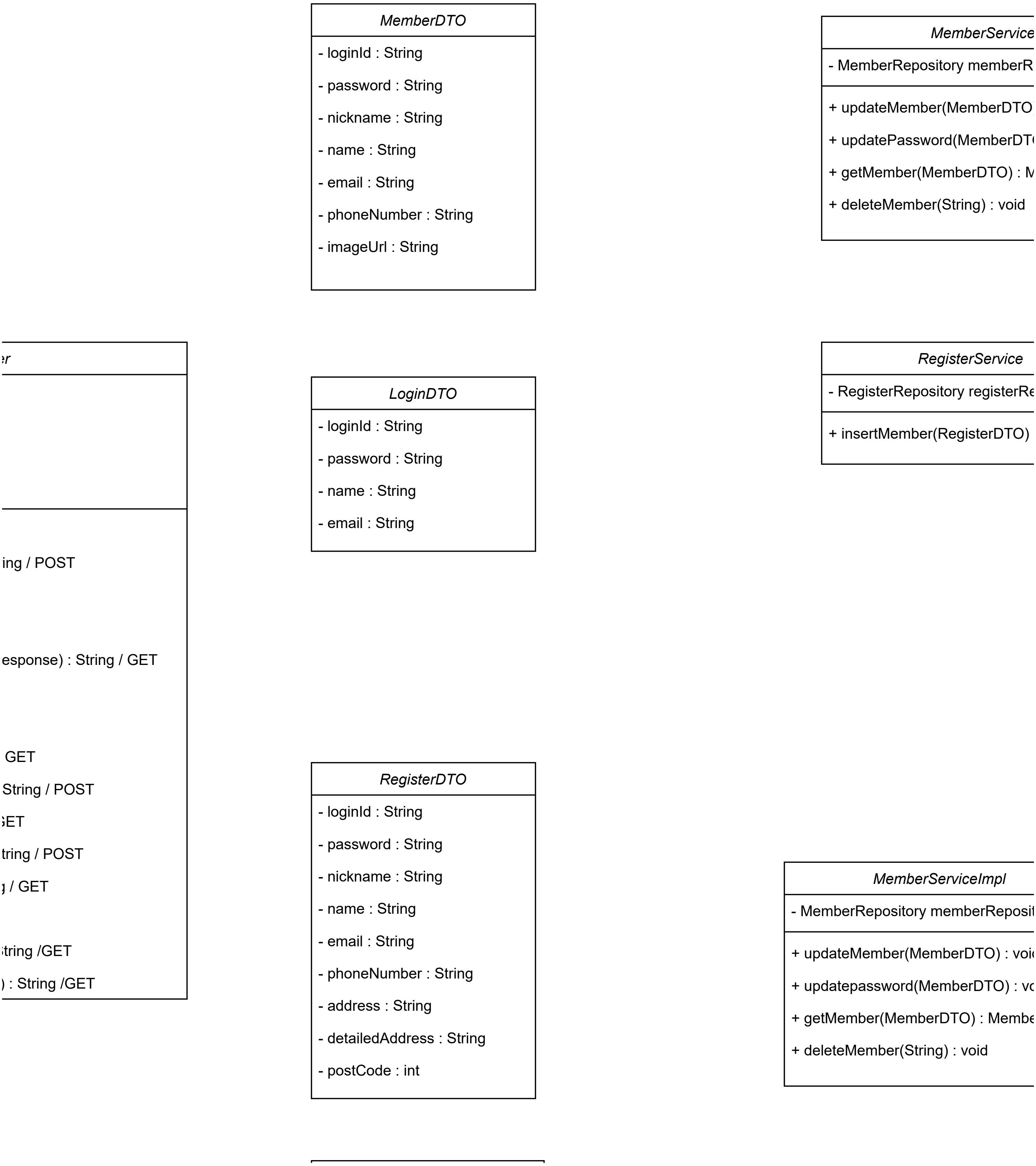

+ save(Delivery delivery): Delivery
Delivery
+ name : String//이름
+ phoneNumber : String//전화번호
+ email : String//이메일
+ address : String//주소
+ detailedAddress : String//상세주소
+ postCode : Integer//우편번호
+ message : String//메세지



MemberControlle

- MemberService member
- LoginService login
- RegisterService register
- OrderMemberService

+ insertMember() : String / GET
+ insertMember(RegisterDTO, Model) : Stri
+ login() : String / GET
+ login() : String / POST
+ logout(HttpServletRequest, HttpServletRe
+ searchId() : String / GET
+ searchId() : String / POST
+ updatePassword(MemberDTO) : String /
+ updatePassword(MemberDTO, Model) : S
+ updateMember(MemberDTO) : String / G
+ updateMember(MemberDTO, Model) : St
+ getMember(Model, MemberDTO) : String
+ deleteMember(String) : String / POST
+ getOrderMember(OrderMemberDTO) : S
+ getOrderMemberList(OrderMemberDTO)



MemberDTO

- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- imageUrl : String

MemberService

- MemberRepository memberR

+ updateMember(MemberDTO
+ updatePassword(MemberDT
+ getMember(MemberDTO) : M
+ deleteMember(String) : void

er

ing / POST

esponse) : String / GET

GET

String / POST

ET

tring / POST

g / GET

tring /GET

) : String /GET

LoginDTO

- loginId : String
- password : String
- name : String
- email : String

RegisterService

- RegisterRepository registerRe

+ insertMember(RegisterDTO)

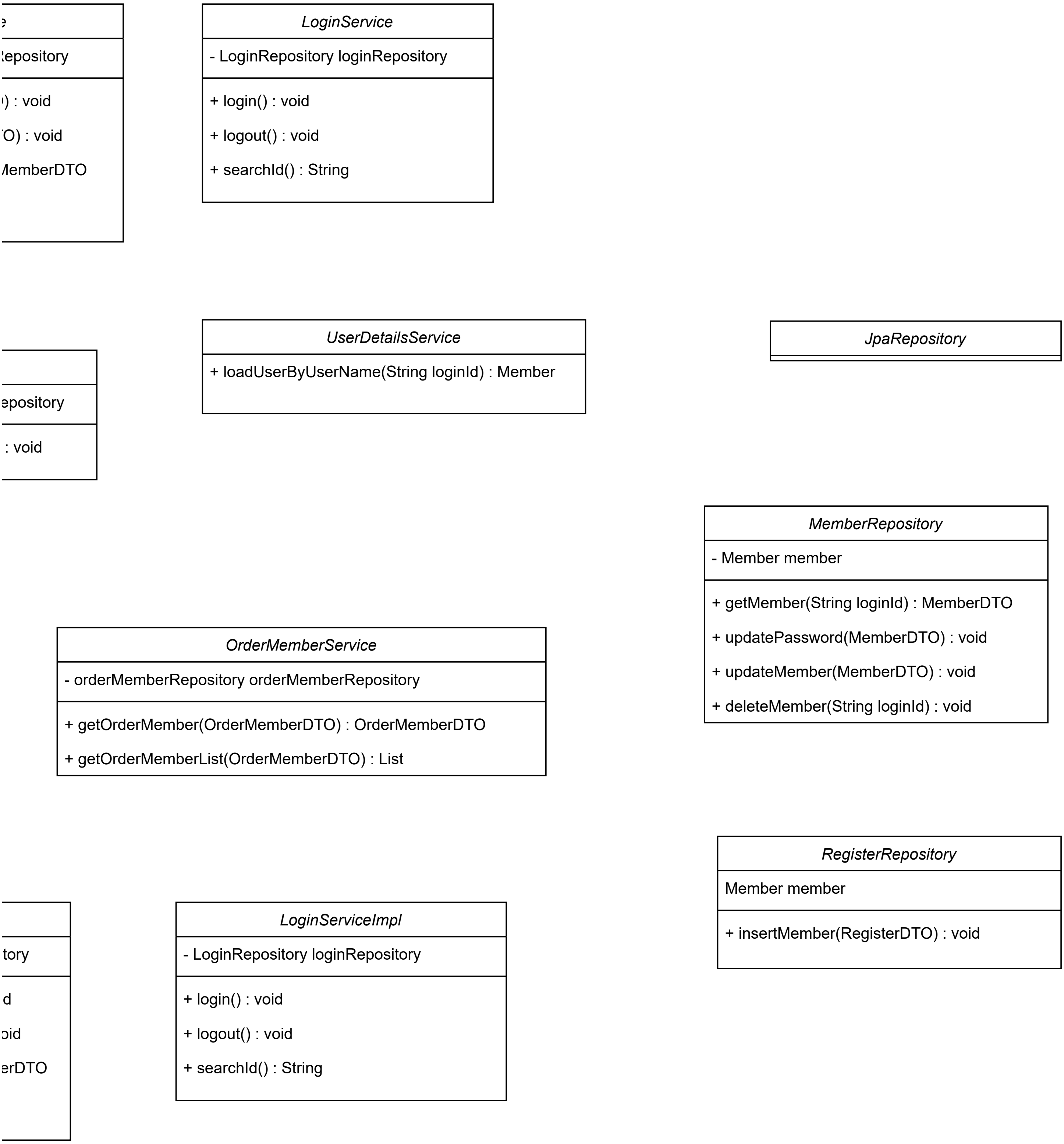
RegisterDTO

- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- address : String
- detailedAddress : String
- postCode : int

MemberServiceImpl

- MemberRepository memberReposit

+ updateMember(MemberDTO) : voi
+ updatepassword(MemberDTO) : vo
+ getMember(MemberDTO) : Membe
+ deleteMember(String) : void



e

:epository

) : void

'O) : void

/lemberDTO

LoginService

- LoginRepository loginRepository

+ login() : void

+ logout() : void

+ searchId() : String

epository

: void

UserDetailsService

+ loadUserByUserName(String loginId) : Member

JpaRepository

MemberRepository

- Member member

+ getMember(String loginId) : MemberDTO

+ updatePassword(MemberDTO) : void

+ updateMember(MemberDTO) : void

+ deleteMember(String loginId) : void

OrderMemberService

- orderMemberRepository orderMemberRepository

+ getOrderMember(OrderMemberDTO) : OrderMemberDTO

+ getOrderMemberList(OrderMemberDTO) : List

RegisterRepository

Member member

+ insertMember(RegisterDTO) : void

tory

d

oid

erDTO

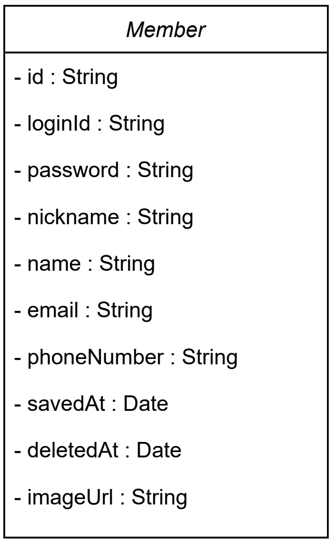
LoginServiceImpl

- LoginRepository loginRepository

+ login() : void

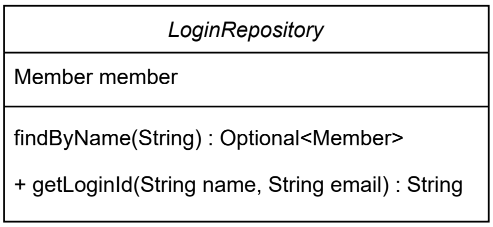
+ logout() : void

+ searchId() : String



Member

- id : String
- loginId : String
- password : String
- nickname : String
- name : String
- email : String
- phoneNumber : String
- savedAt : Date
- deletedAt : Date
- imageUrl : String

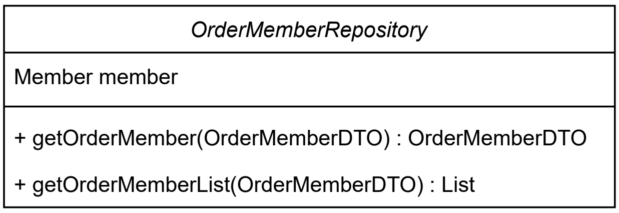


LoginRepository

Member member

findByName(String) : Optional<Member>

+ getLoginId(String name, String email) : String

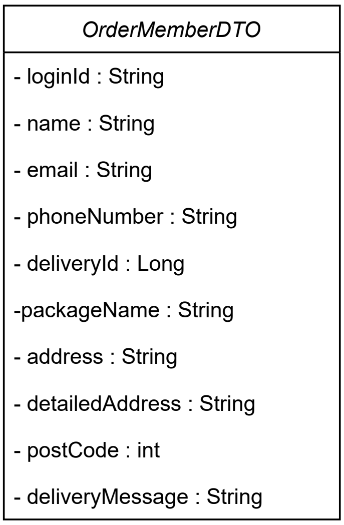


OrderMemberRepository

Member member

+ getOrderMember(OrderMemberDTO) : OrderMemberDTO

+ getOrderMemberList(OrderMemberDTO) : List

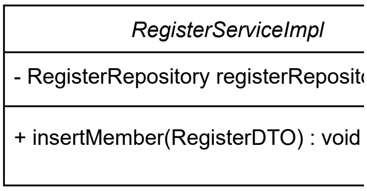


OrderMemberDTO

- loginId : String
- name : String
- email : String
- phoneNumber : String
- deliveryId : Long

-packageName : String

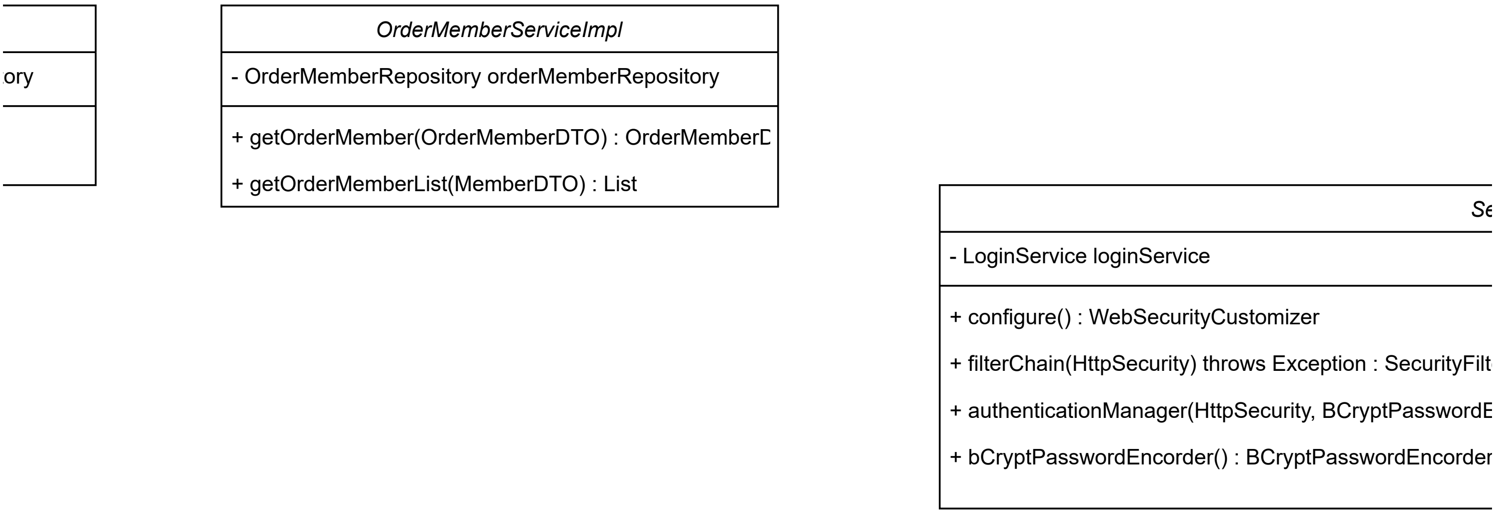
- address : String
- detailedAddress : String
- postCode : int
- deliveryMessage : String

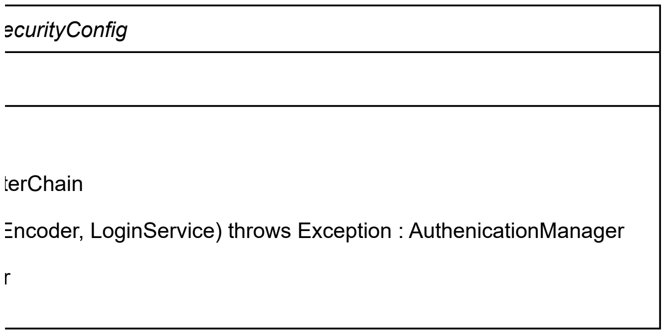


RegisterServiceImpl

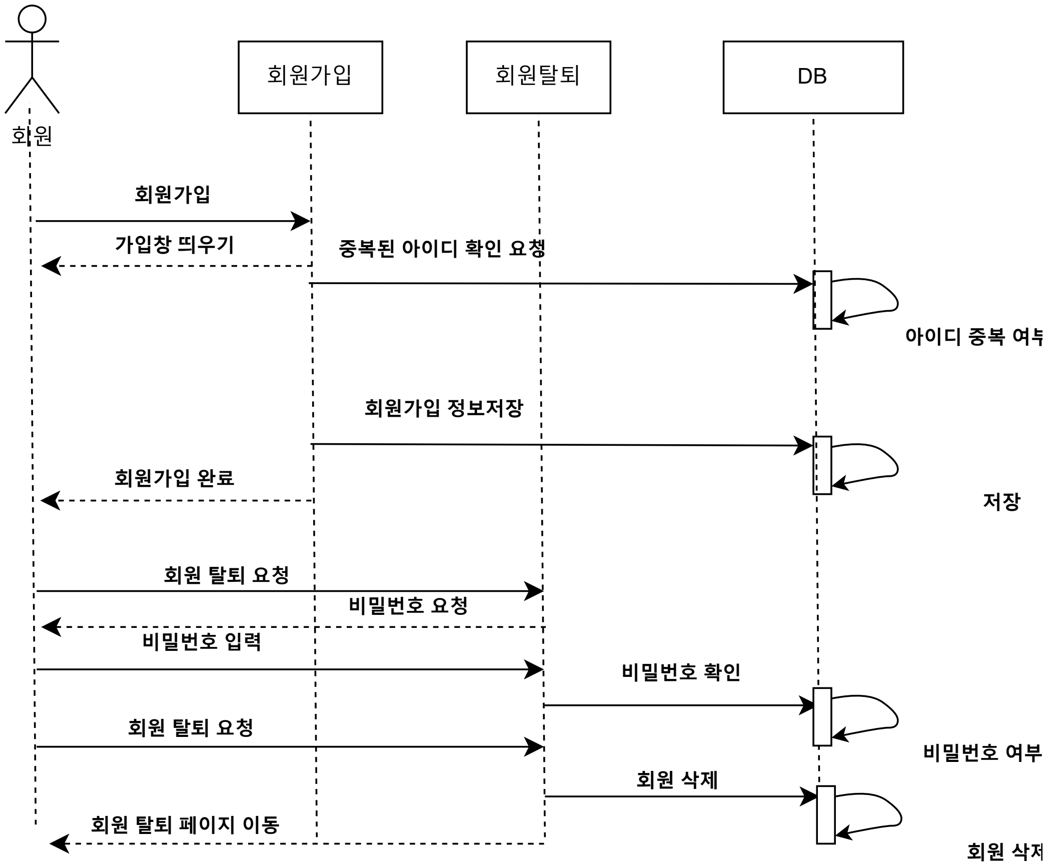
- RegisterRepository registerReposit

+ insertMember(RegisterDTO) : void

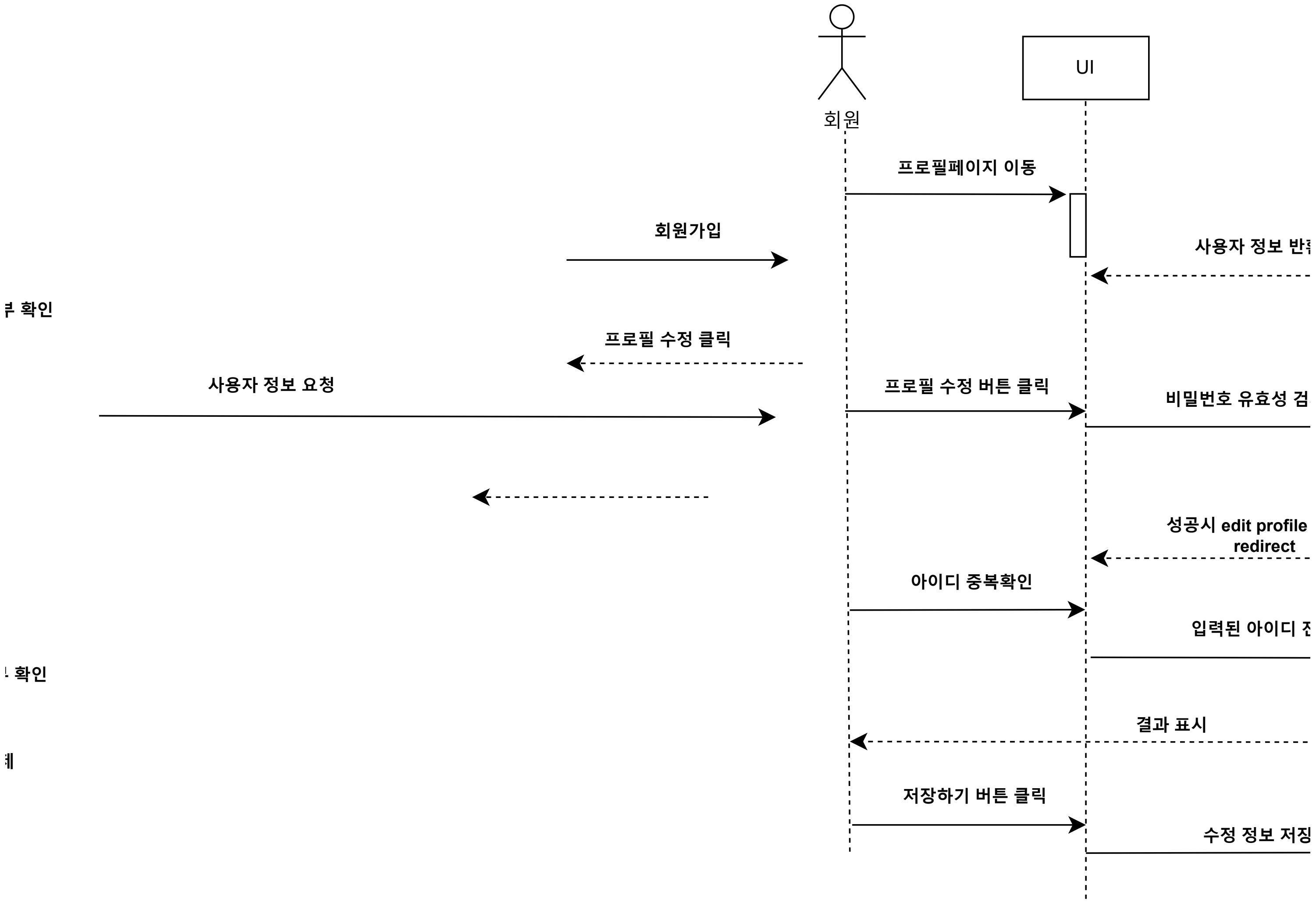

ory
OrderMemberServiceImpl
- OrderMemberRepository orderMemberRepository
+ getOrderMember(OrderMemberDTO) : OrderMemberD
+ getOrderMemberList(MemberDTO) : List
Se
- LoginService loginService
+ configure() : WebSecurityCustomizer
+ filterChain(HttpSecurity) throws Exception : SecurityFilt
+ authenticationManager(HttpSecurity, BCryptPasswordE
+ bCryptPasswordEncorder() : BCryptPasswordEncorder

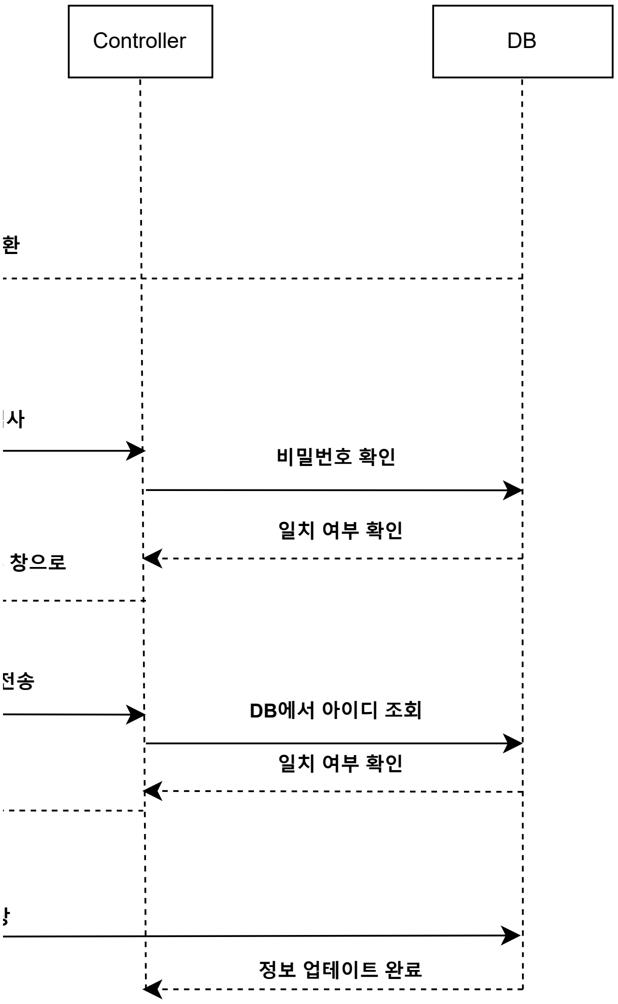

ecurityConfig
terChain
Encoder, LoginService) throws Exception : AuthenicationManager
r

회원가입

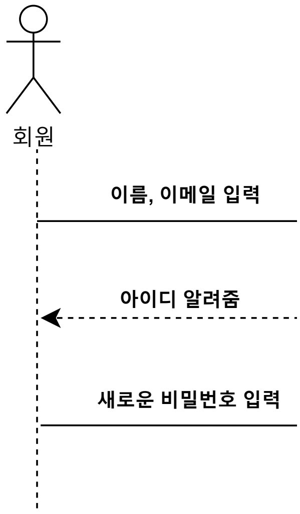

회원
회원가입
회원탈퇴
DB
회원가입
가입창 띄우기
중복된 아이디 확인 요청
아이디 중복 여부
회원가입 정보저장
회원가입 완료
저장
회원 탈퇴 요청
비밀번호 요청
비밀번호 입력
비밀번호 확인
회원 탈퇴 요청
비밀번호 여부
회원 삭제
회원 탈퇴 페이지 이동
회원 삭제

프로필

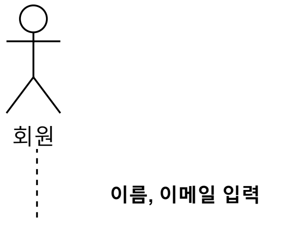

회원
UI
프로필페이지 이동
회원가입
사용자 정보 반
확인
프로필 수정 클릭
사용자 정보 요청
프로필 수정 버튼 클릭
비밀번호 유효성 검
성공시 edit profile
redirect
아이디 중복확인
입력된 아이디
확인
결과 표시
저장하기 버튼 클릭
수정 정보 저장


Controller
DB
환
사
비밀번호 확인
일치 여부 확인
창으로
전송
DB에서 아이디 조회
일치 여부 확인
정보 업데이트 완료

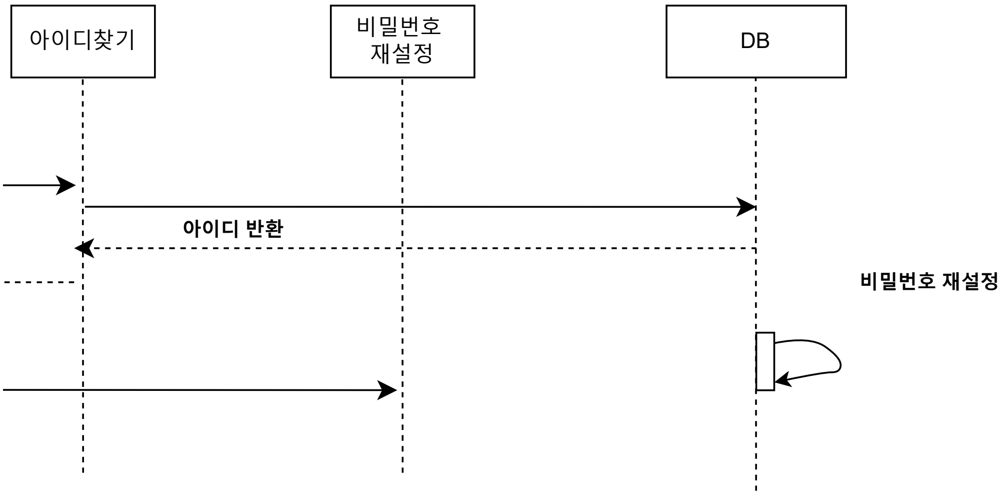
프로젝트 등


회원
이름, 이메일 입력
아이디 알려줌
새로운 비밀번호 입력

후원하

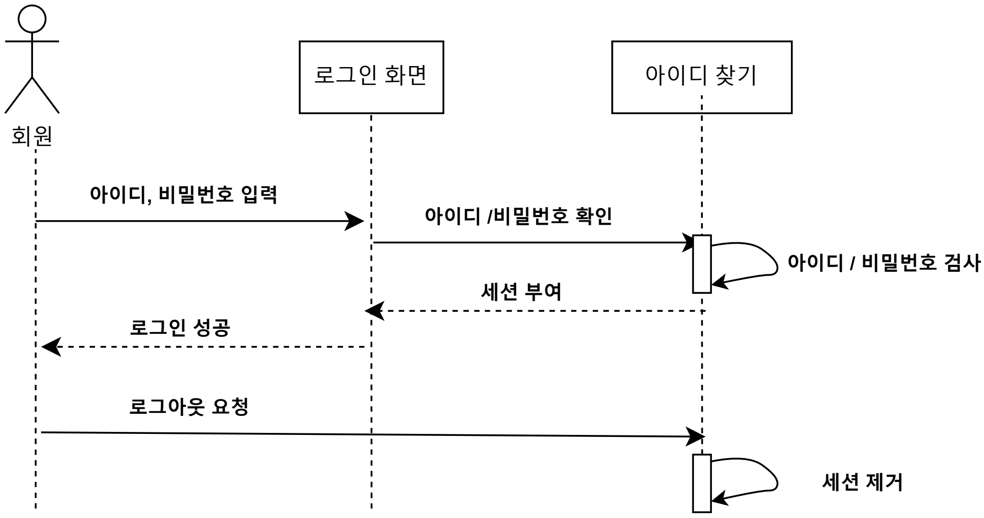

회원
이름, 이메일 입력

록하기

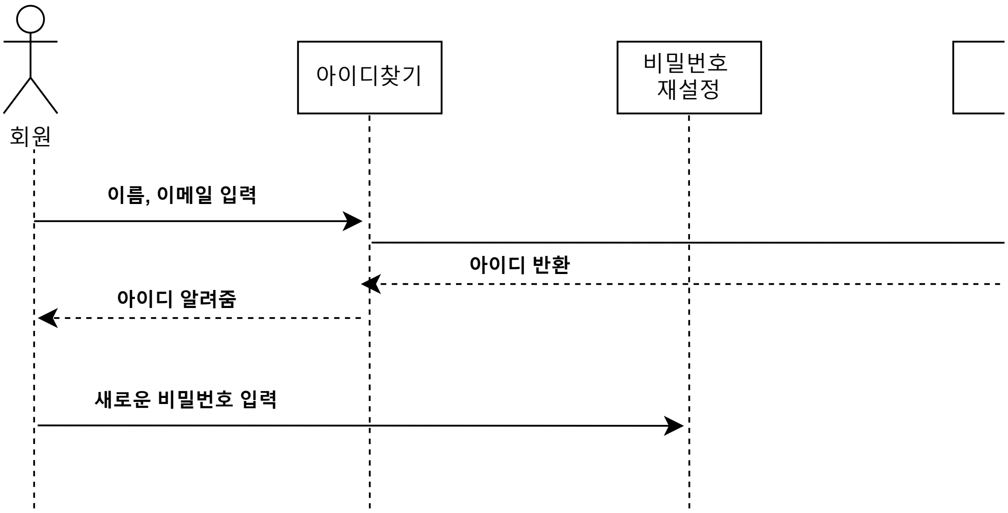


기

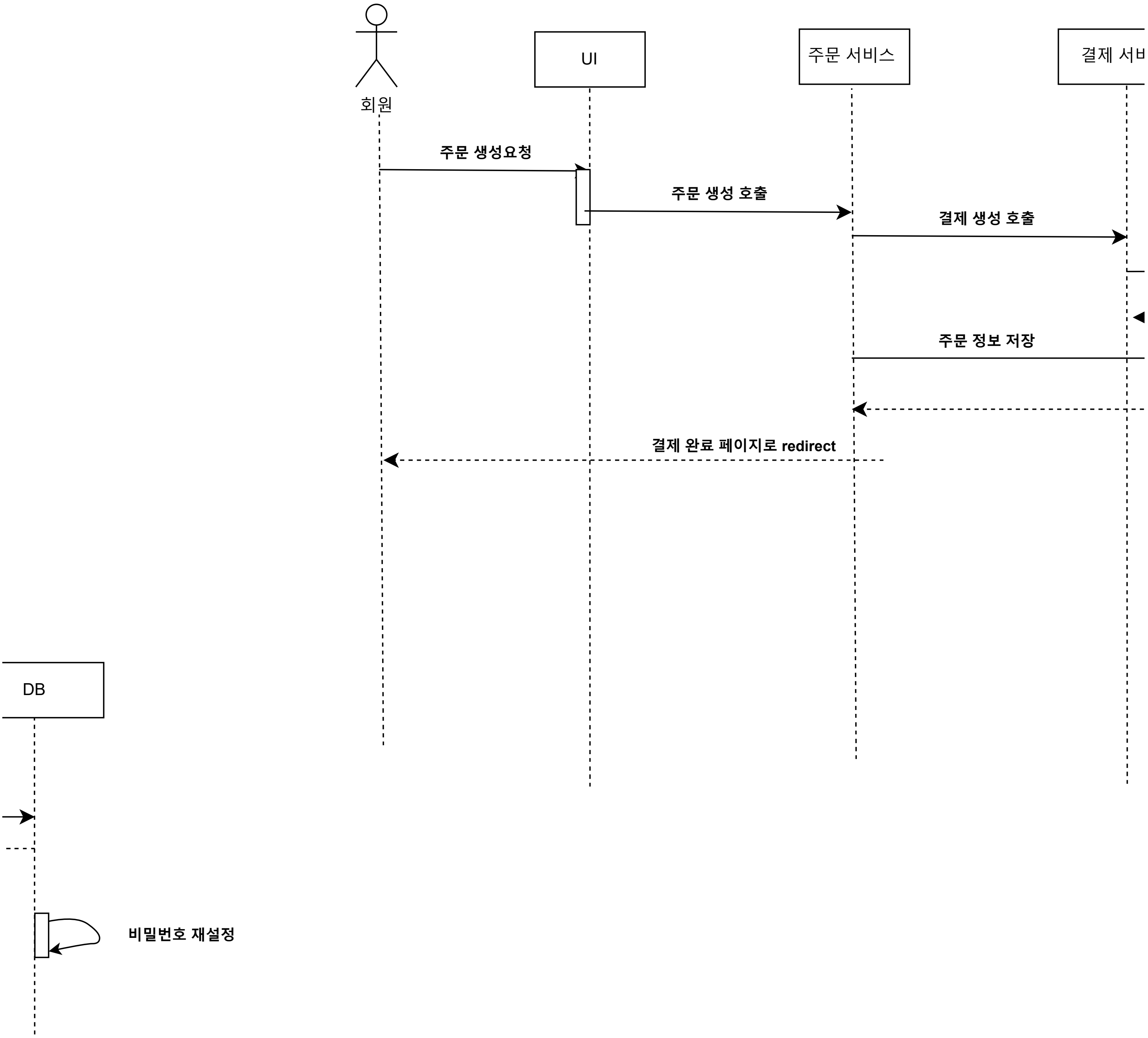
로그인

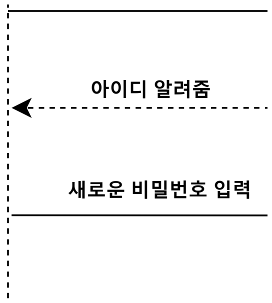


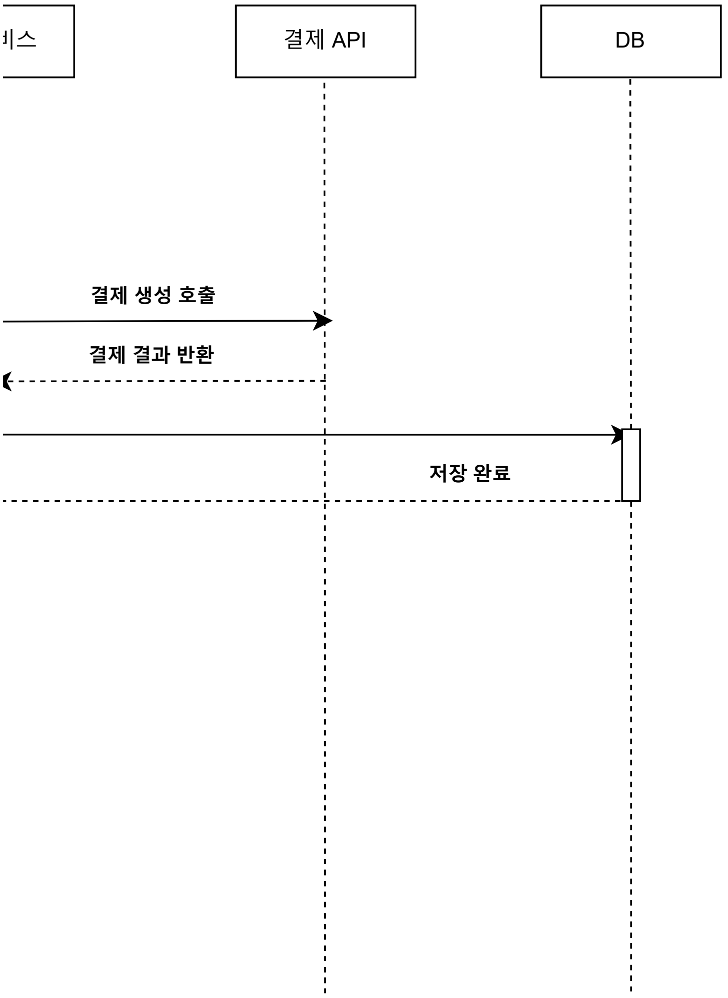
아이디찾기/ 비밀번호 재설정

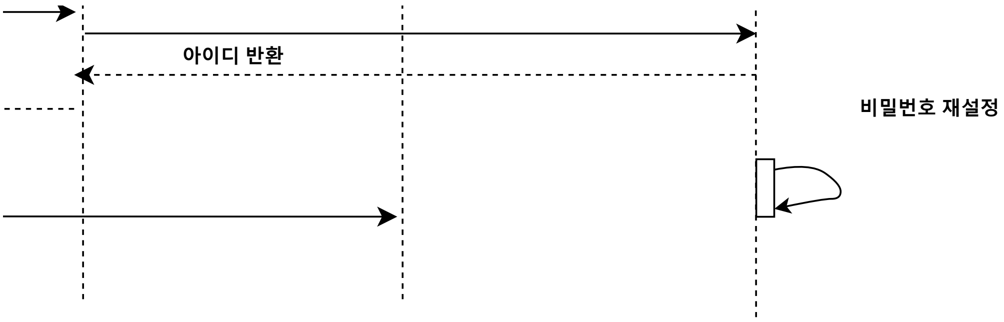


주문 &결제


회원
UI
주문 서비스
결제 서ㅂ
주문 생성요청
주문 생성 호출
결제 생성 호출
주문 정보 저장
결제 완료 페이지로 redirect
DB
비밀번호 재설정


아이디 알려줌
새로운 비밀번호 입력


비스
결제 API
DB
결제 생성 호출
결제 결과 반환
저장 완료


아이디 반환
비밀번호 재설정

이거

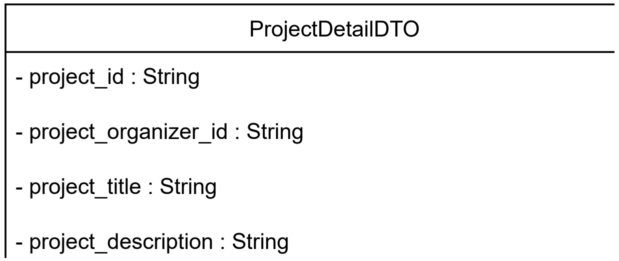
살려줘

난 아무것도 몰라

이게뭐야

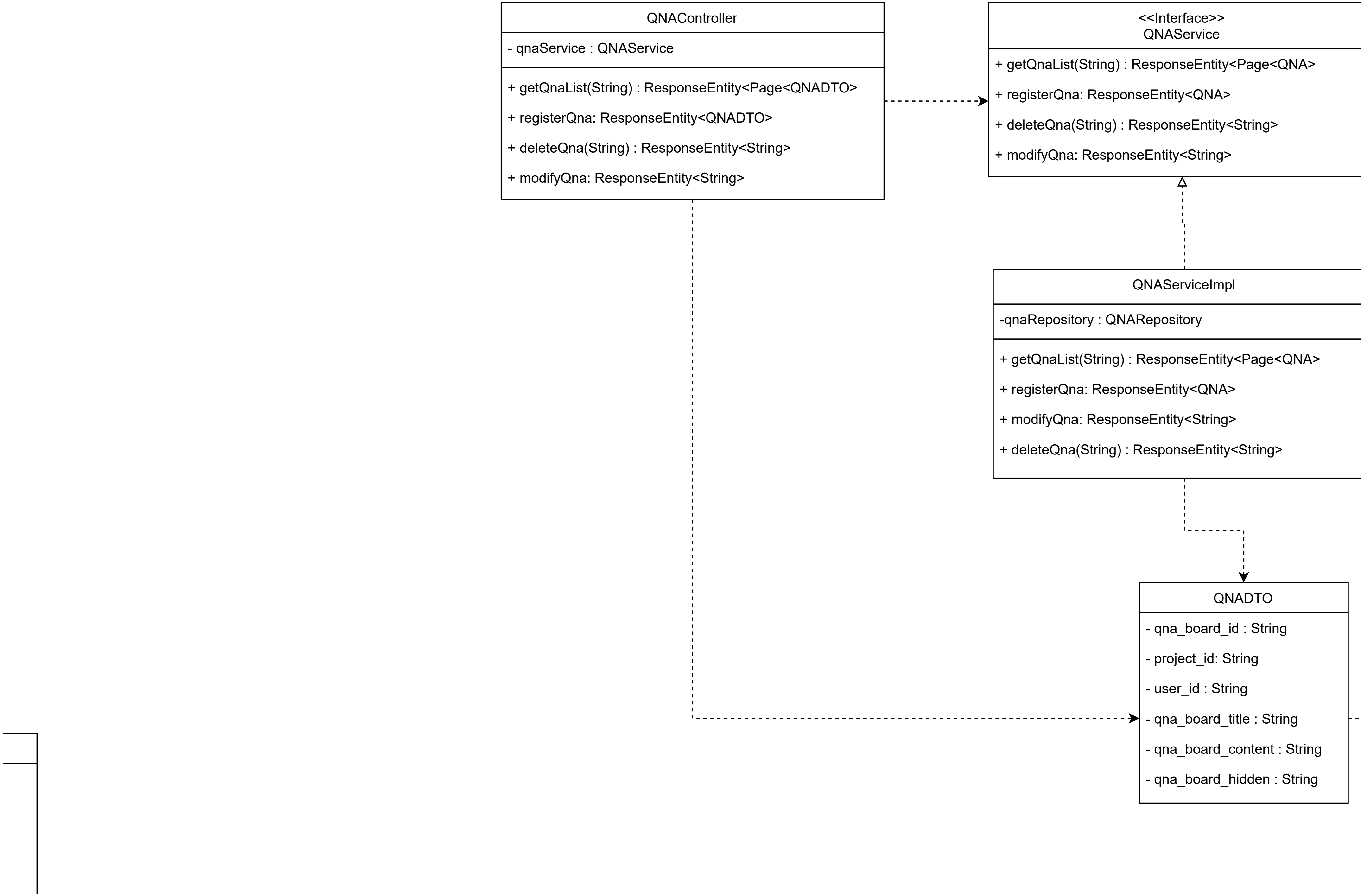
너무 귀여운거 아닌가요 - 주현

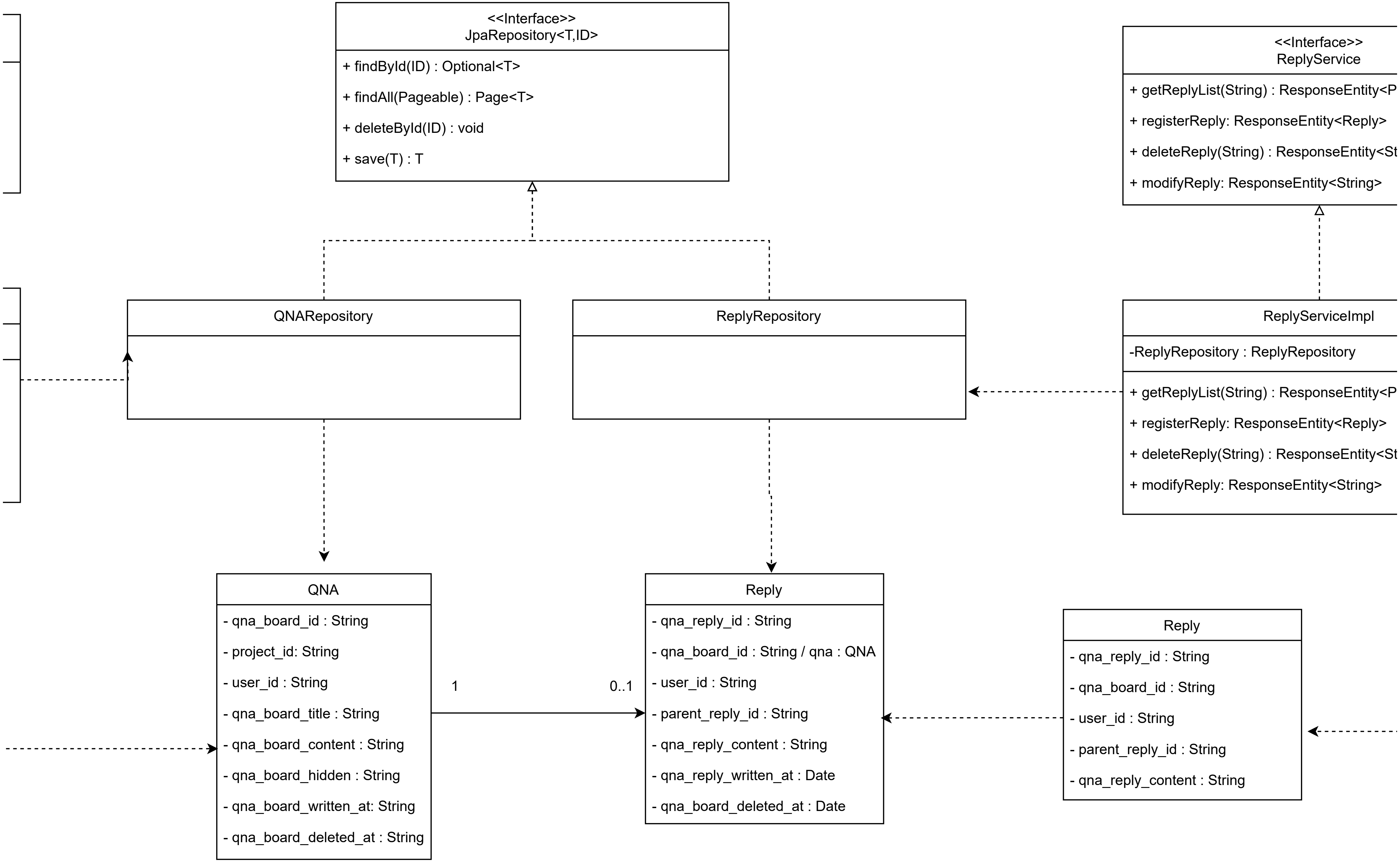
흑흑.. ❤

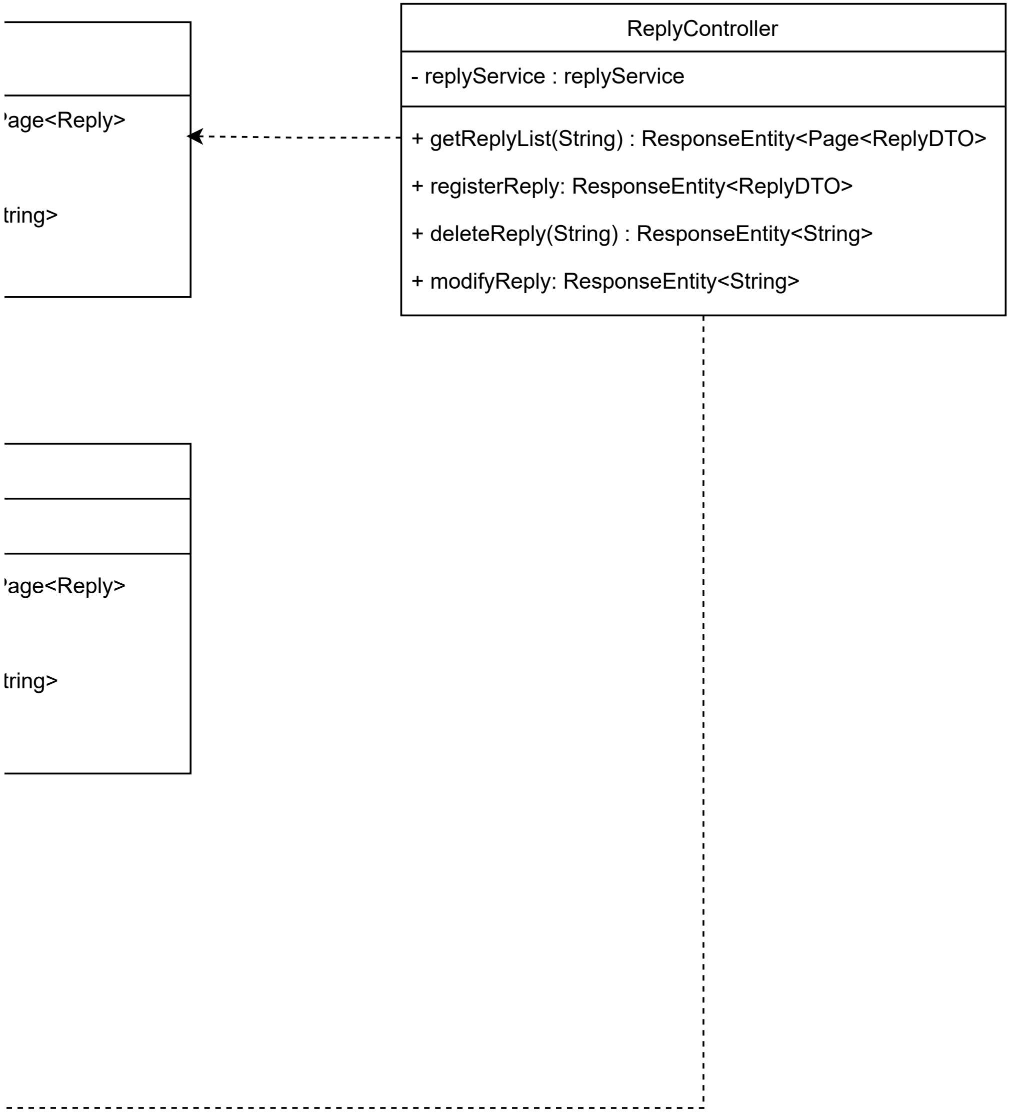


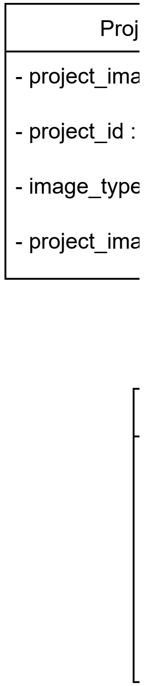
ProjectDetailDTO

- project_id : String
- project_organizer_id : String
- project_title : String
- project_description : String


QNAController
- qnaService : QNAService
+ getQnaList(String) : ResponseEntity<Page<QNADTO>
+ registerQna: ResponseEntity<QNADTO>
+ deleteQna(String) : ResponseEntity<String>
+ modifyQna: ResponseEntity<String>
<<Interface>>
QNAService
+ getQnaList(String) : ResponseEntity<Page<QNA>
+ registerQna: ResponseEntity<QNA>
+ deleteQna(String) : ResponseEntity<String>
+ modifyQna: ResponseEntity<String>
QNAServiceImpl
-qnaRepository : QNARepository
+ getQnaList(String) : ResponseEntity<Page<QNA>
+ registerQna: ResponseEntity<QNA>
+ modifyQna: ResponseEntity<String>
+ deleteQna(String) : ResponseEntity<String>
QNADTO
- qna_board_id : String
- project_id: String
- user_id : String
- qna_board_title : String
- qna_board_content : String
- qna_board_hidden : String


<<Interface>>
JpaRepository<T,ID>
+ findById(ID) : Optional<T>
+ findAll(Pageable) : Page<T>
+ deleteById(ID) : void
+ save(T) : T
<<Interface>>
ReplyService
+ getReplyList(String) : ResponseEntity<P
+ registerReply: ResponseEntity<Reply>
+ deleteReply(String) : ResponseEntity<St
+ modifyReply: ResponseEntity<String>
QNARepository
ReplyRepository
ReplyServiceImpl
-ReplyRepository : ReplyRepository
+ getReplyList(String) : ResponseEntity<P
+ registerReply: ResponseEntity<Reply>
+ deleteReply(String) : ResponseEntity<St
+ modifyReply: ResponseEntity<String>
QNA
- qna_board_id : String
- project_id: String
- user_id : String
- qna_board_title : String
- qna_board_content : String
- qna_board_hidden : String
- qna_board_written_at: String
- qna_board_deleted_at : String
1
0..1
Reply
- qna_reply_id : String
- qna_board_id : String / qna : QNA
- user_id : String
- parent_reply_id : String
- qna_reply_content : String
- qna_reply_written_at : Date
- qna_board_deleted_at : Date
Reply
- qna_reply_id : String
- qna_board_id : String
- user_id : String
- parent_reply_id : String
- qna_reply_content : String


'age<Reply>
tring>
'age<Reply>
tring>
ReplyController
- replyService : replyService
+ getReplyList(String) : ResponseEntity<Page<ReplyDTO>
+ registerReply: ResponseEntity<ReplyDTO>
+ deleteReply(String) : ResponseEntity<String>
+ modifyReply: ResponseEntity<String>



Proj

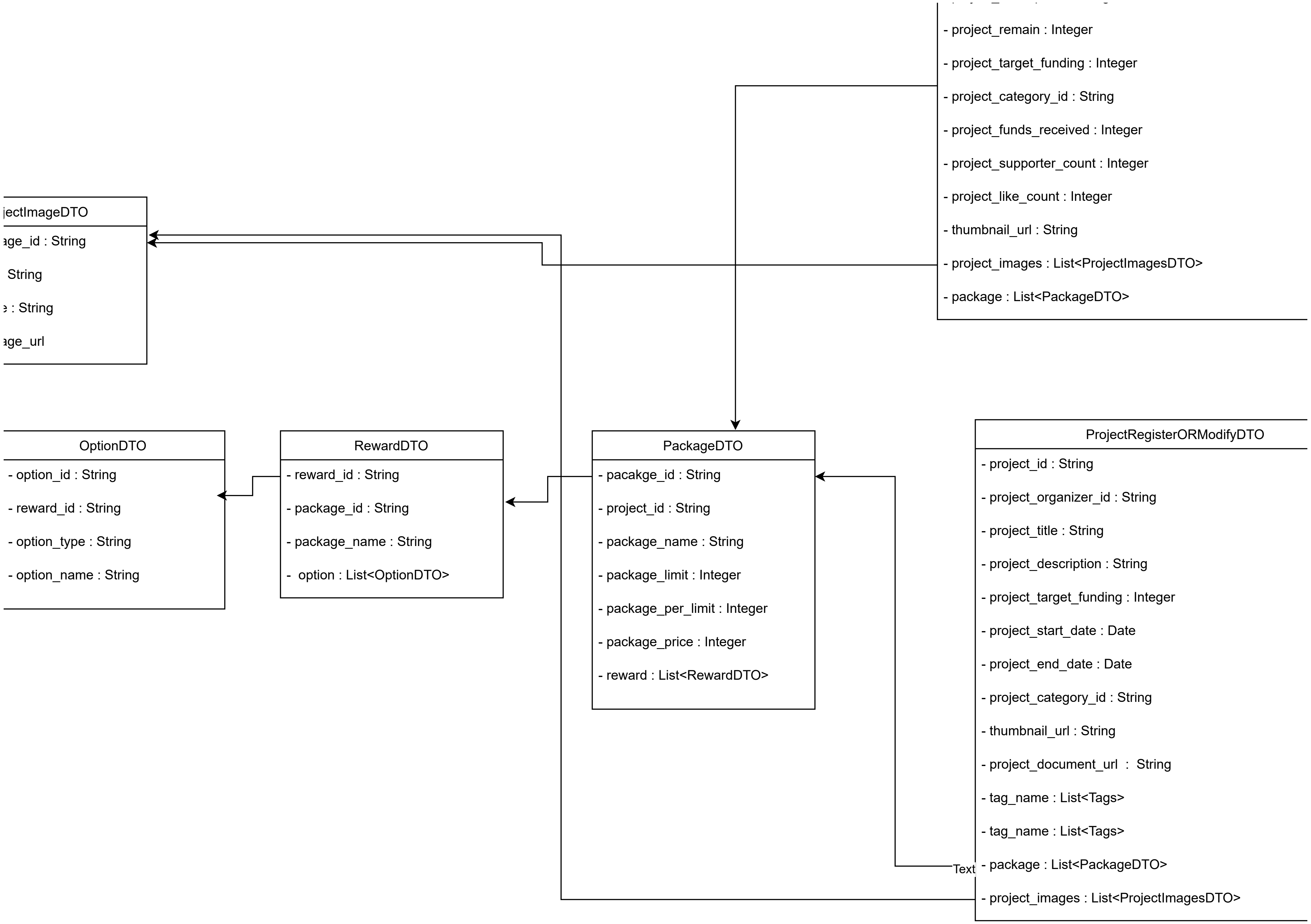
- project_ima

- project_id :

- image_type

- project_ima

Repository 만들기(고민하기)



- project_remain : Integer
- project_target_funding : Integer
- project_category_id : String
- project_funds_received : Integer
- project_supporter_count : Integer
- project_like_count : Integer
- thumbnail_url : String
- project_images : List<ProjectImagesDTO>
- package : List<PackageDTO>

jectImageDTO

age_id : String

String

: String

age_url

OptionDTO

- option_id : String
- reward_id : String
- option_type : String
- option_name : String

RewardDTO

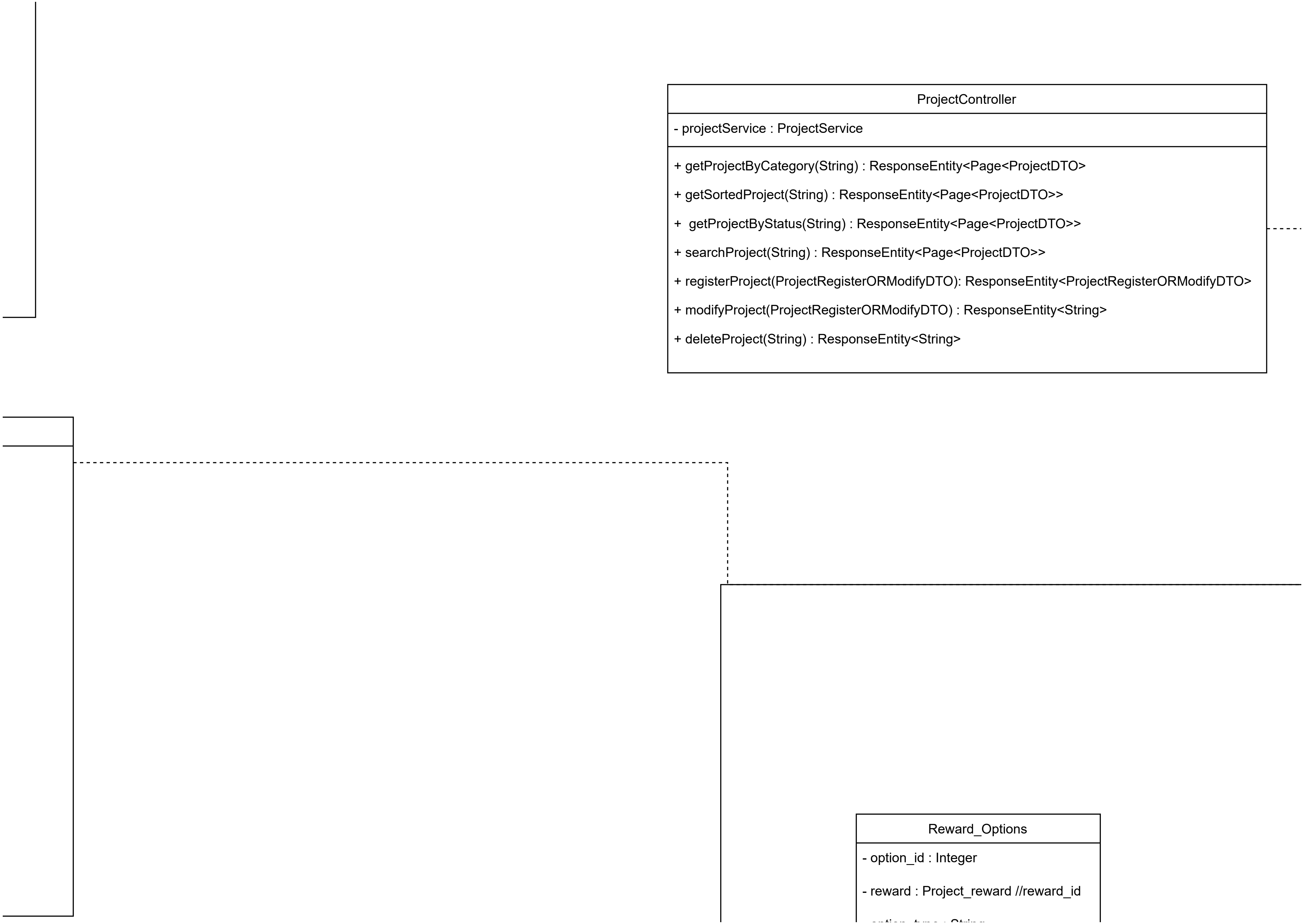
- reward_id : String
- package_id : String
- package_name : String
- option : List<OptionDTO>

PackageDTO

- pacakge_id : String
- project_id : String
- package_name : String
- package_limit : Integer
- package_per_limit : Integer
- package_price : Integer
- reward : List<RewardDTO>

ProjectRegisterORModifyDTO

- project_id : String
- project_organizer_id : String
- project_title : String
- project_description : String
- project_target_funding : Integer
- project_start_date : Date
- project_end_date : Date
- project_category_id : String
- thumbnail_url : String
- project_document_url : String
- tag_name : List<Tags>
- tag_name : List<Tags>
- package : List<PackageDTO>
- project_images : List<ProjectImagesDTO>



ProjectController

- projectService : ProjectService

+ getProjectByCategory(String) : ResponseEntity<Page<ProjectDTO>

+ getSortedProject(String) : ResponseEntity<Page<ProjectDTO>>

+ getProjectByStatus(String) : ResponseEntity<Page<ProjectDTO>>

+ searchProject(String) : ResponseEntity<Page<ProjectDTO>>

+ registerProject(ProjectRegisterORModifyDTO): ResponseEntity<ProjectRegisterORModifyDTO>

+ modifyProject(ProjectRegisterORModifyDTO) : ResponseEntity<String>

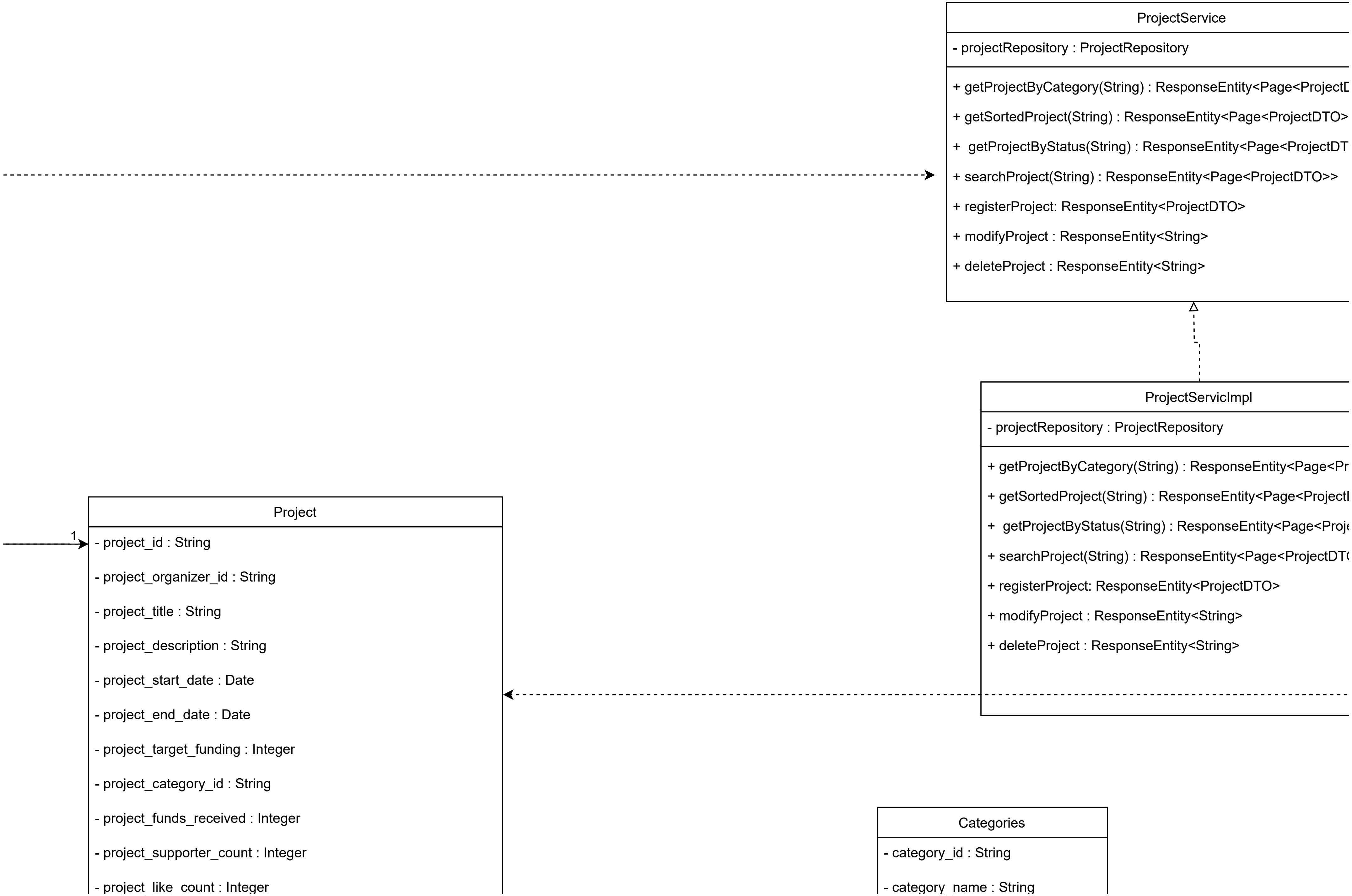
+ deleteProject(String) : ResponseEntity<String>

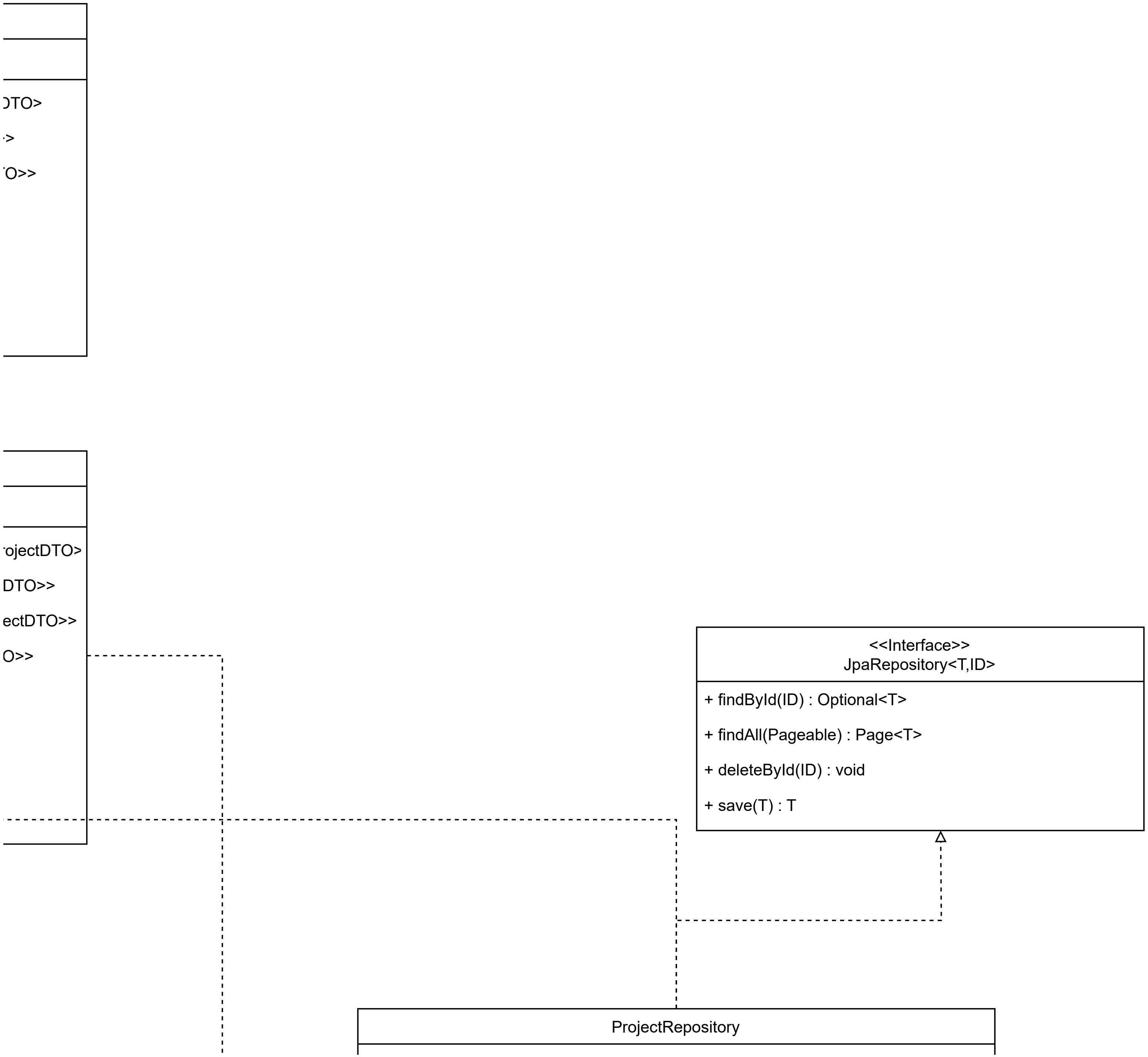
Reward_Options

- option_id : Integer

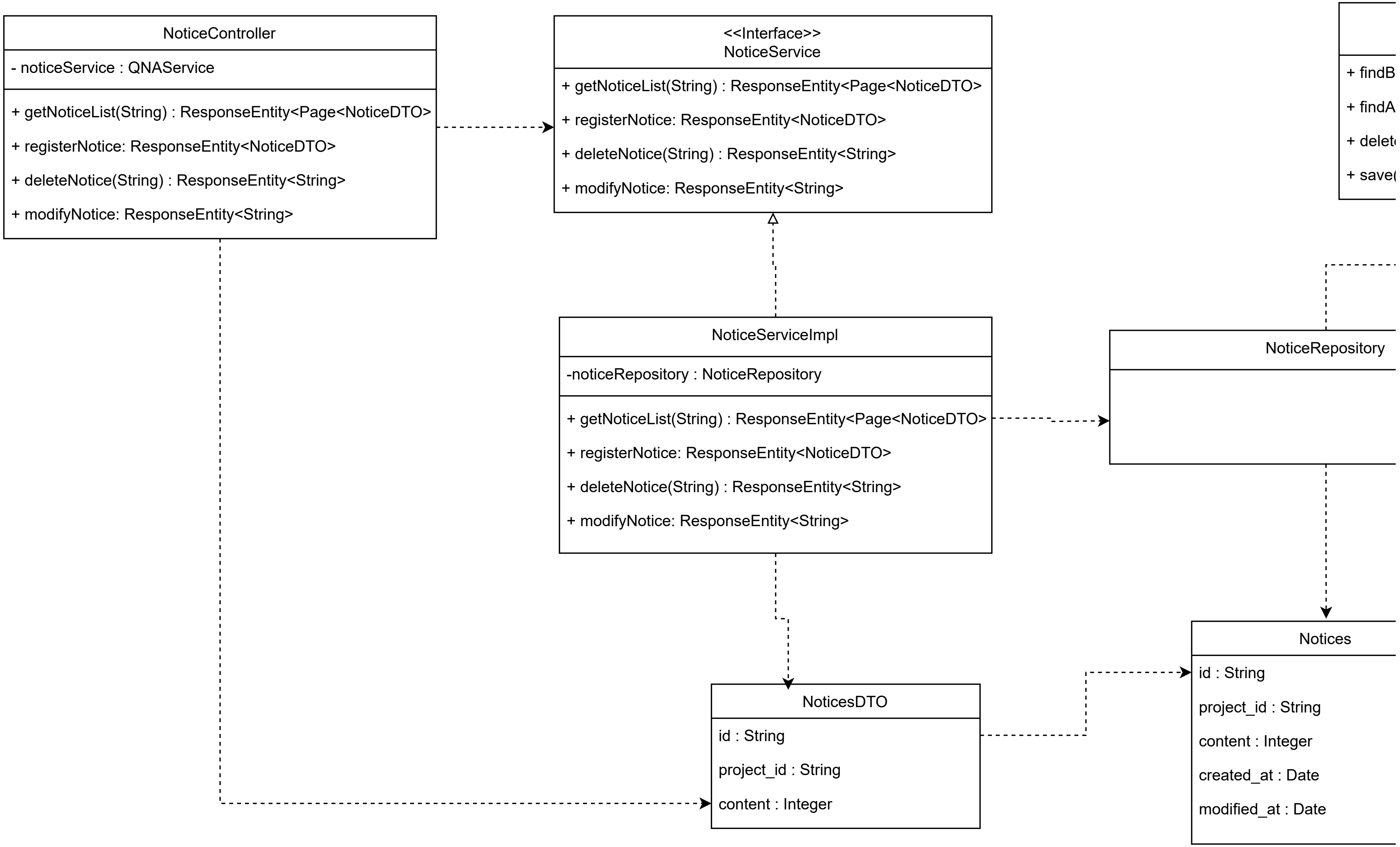
- reward : Project_reward //reward_id

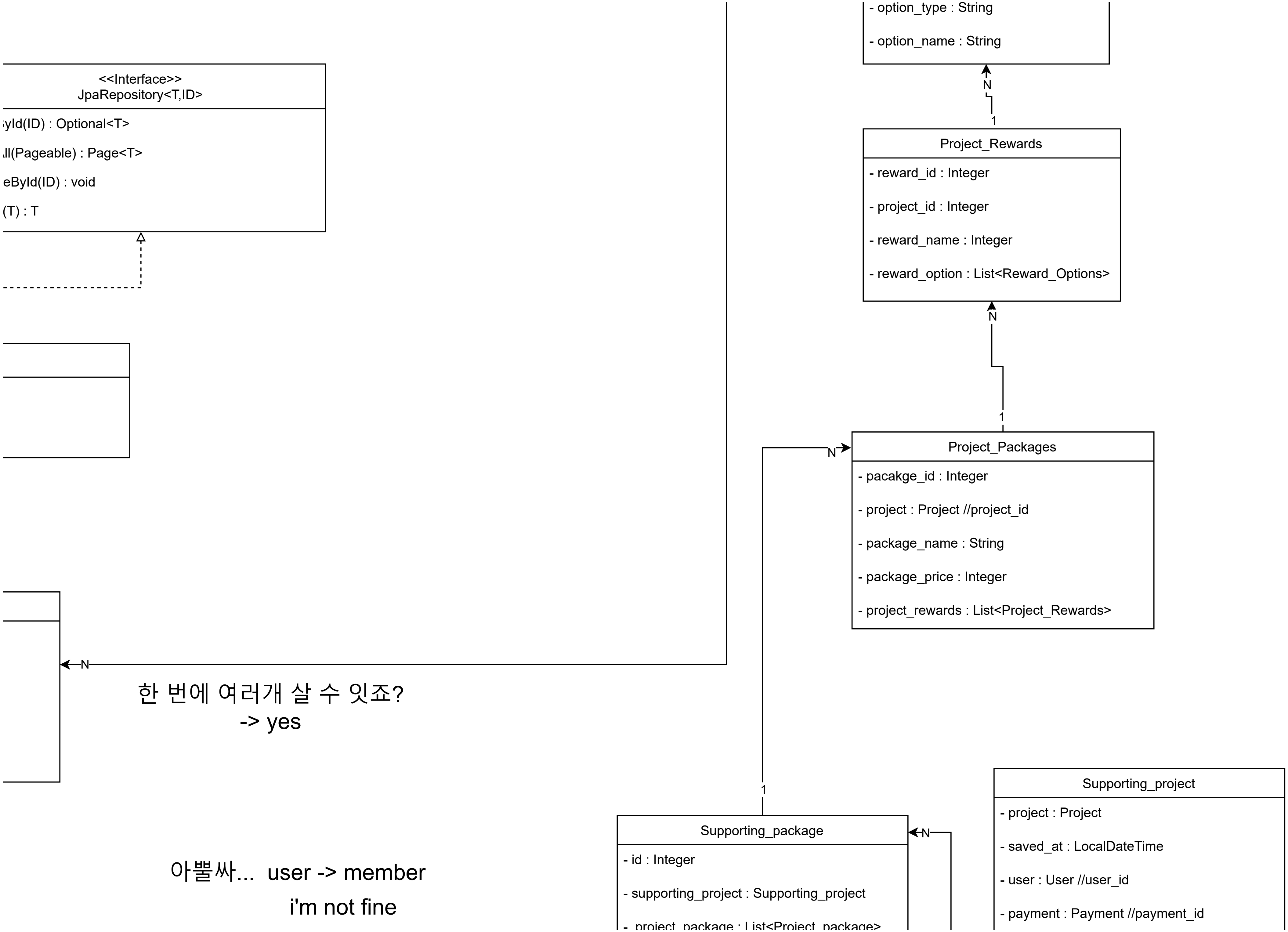
- option_type : String


ProjectService
- projectRepository : ProjectRepository
+ getProjectByCategory(String) : ResponseEntity<Page<ProjectD
+ getSortedProject(String) : ResponseEntity<Page<ProjectDTO>
+ getProjectByStatus(String) : ResponseEntity<Page<ProjectDT
+ searchProject(String) : ResponseEntity<Page<ProjectDTO>>
+ registerProject: ResponseEntity<ProjectDTO>
+ modifyProject : ResponseEntity<String>
+ deleteProject : ResponseEntity<String>
ProjectServicImpl
- projectRepository : ProjectRepository
+ getProjectByCategory(String) : ResponseEntity<Page<Pr
+ getSortedProject(String) : ResponseEntity<Page<ProjectI
+ getProjectByStatus(String) : ResponseEntity<Page<Proje
+ searchProject(String) : ResponseEntity<Page<ProjectDT(
+ registerProject: ResponseEntity<ProjectDTO>
+ modifyProject : ResponseEntity<String>
+ deleteProject : ResponseEntity<String>
1
Project
- project_id : String
- project_organizer_id : String
- project_title : String
- project_description : String
- project_start_date : Date
- project_end_date : Date
- project_target_funding : Integer
- project_category_id : String
- project_funds_received : Integer
- project_supporter_count : Integer
- project_like_count : Integer
Categories
- category_id : String
- category_name : String


)TO>
->
'O>>
·ojectDTO>
DTO>>
ectDTO>>
O>>
<<Interface>>
JpaRepository<T,ID>
+ findById(ID) : Optional<T>
+ findAll(Pageable) : Page<T>
+ deleteById(ID) : void
+ save(T) : T
ProjectRepository

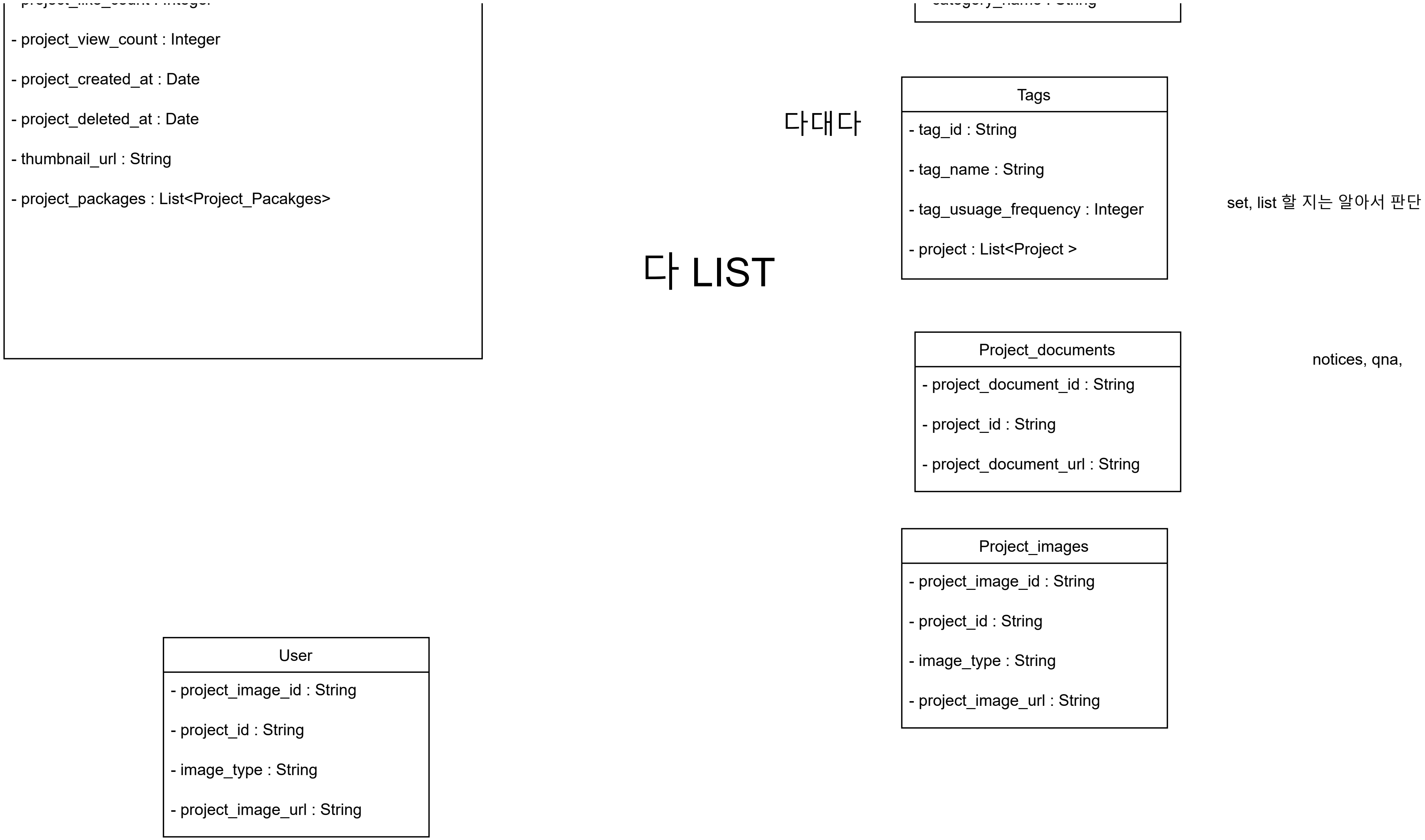
변수명 규칙->


NoticeController
- noticeService : QNAService
+ getNoticeList(String) : ResponseEntity<Page<NoticeDTO>
+ registerNotice: ResponseEntity<NoticeDTO>
+ deleteNotice(String) : ResponseEntity<String>
+ modifyNotice: ResponseEntity<String>
<<Interface>>
NoticeService
+ getNoticeList(String) : ResponseEntity<Page<NoticeDTO>
+ registerNotice: ResponseEntity<NoticeDTO>
+ deleteNotice(String) : ResponseEntity<String>
+ modifyNotice: ResponseEntity<String>
+ findB
+ findA
+ delet
+ save(
NoticeServiceImpl
-noticeRepository : NoticeRepository
+ getNoticeList(String) : ResponseEntity<Page<NoticeDTO>
+ registerNotice: ResponseEntity<NoticeDTO>
+ deleteNotice(String) : ResponseEntity<String>
+ modifyNotice: ResponseEntity<String>
NoticeRepository
Notices
id : String
project_id : String
content : Integer
created_at : Date
modified_at : Date
NoticesDTO
id : String
project_id : String
content : Integer



한 번에 여러개 살 수 잇죠?
-> yes

아뿔싸... user -> member
i'm not fine



- project_like_count : Integer

- project_view_count : Integer

- project_created_at : Date

- project_deleted_at : Date

- thumbnail_url : String

- project_packages : List<Project_Pacakges>

category_name : String

다대다

Tags

- tag_id : String

- tag_name : String

- tag_usuage_frequency : Integer

- project : List<Project >

set, list 할 지는 알아서 판단

다 LIST

Project_documents

- project_document_id : String

- project_id : String

- project_document_url : String

notices, qna,

Project_images

- project_image_id : String

- project_id : String

- image_type : String

- project_image_url : String

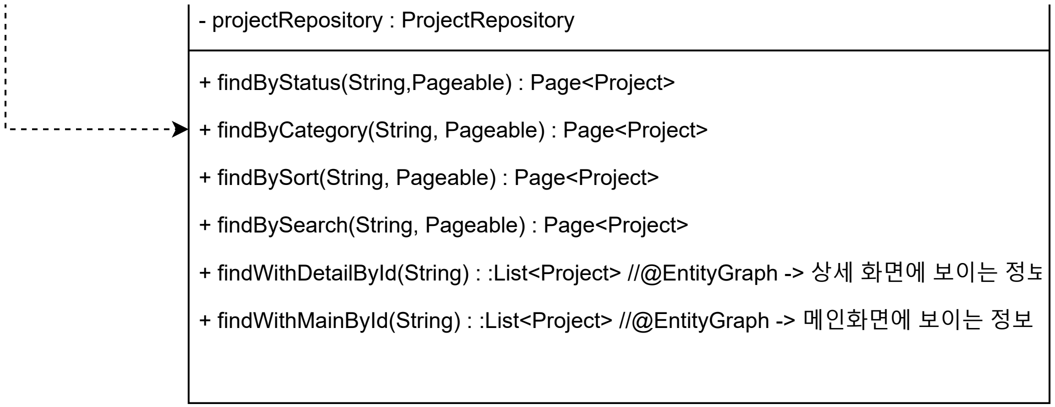
User

- project_image_id : String

- project_id : String

- image_type : String

- project_image_url : String



- projectRepository : ProjectRepository

+ findByStatus(String,Pageable) : Page<Project>

+ findByCategory(String, Pageable) : Page<Project>

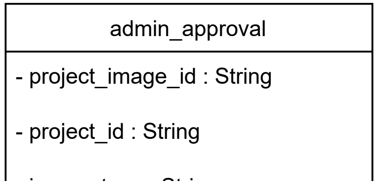
+ findBySort(String, Pageable) : Page<Project>

+ findBySearch(String, Pageable) : Page<Project>

+ findWithDetailById(String) : :List<Project> //@EntityGraph -> 상세 화면에 보이는 정보

+ findWithMainById(String) : :List<Project> //@EntityGraph -> 메인화면에 보이는 정보

태욱님...것이라...믿습니다..



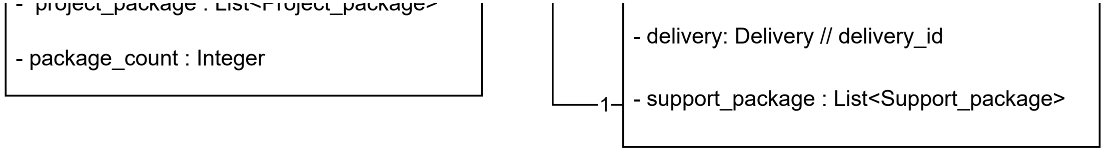
admin_approval

- project_image_id : String

- project_id : String

datetime,,, 을

spring boot어
JoinColumn n



- project_package : List<Project_package>

- package_count : Integer

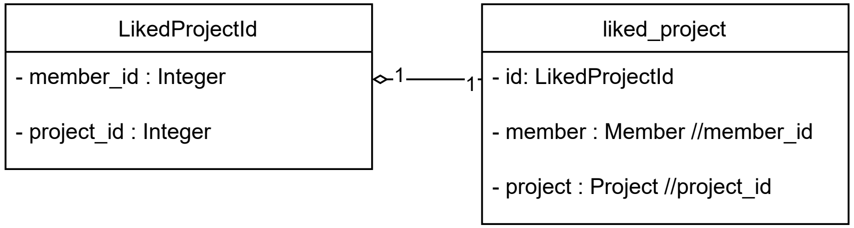
- delivery: Delivery // delivery_id

- support_package : List<Support_package>

뭐라고,,,,하더라,,,?

ㅔ서는 연관관계를
ame으로 하던데,,,,

복합키 구성을 위한 클래스.
@EmbeddedId
@MapsId



LikedProjectId

- member_id : Integer

- project_id : Integer

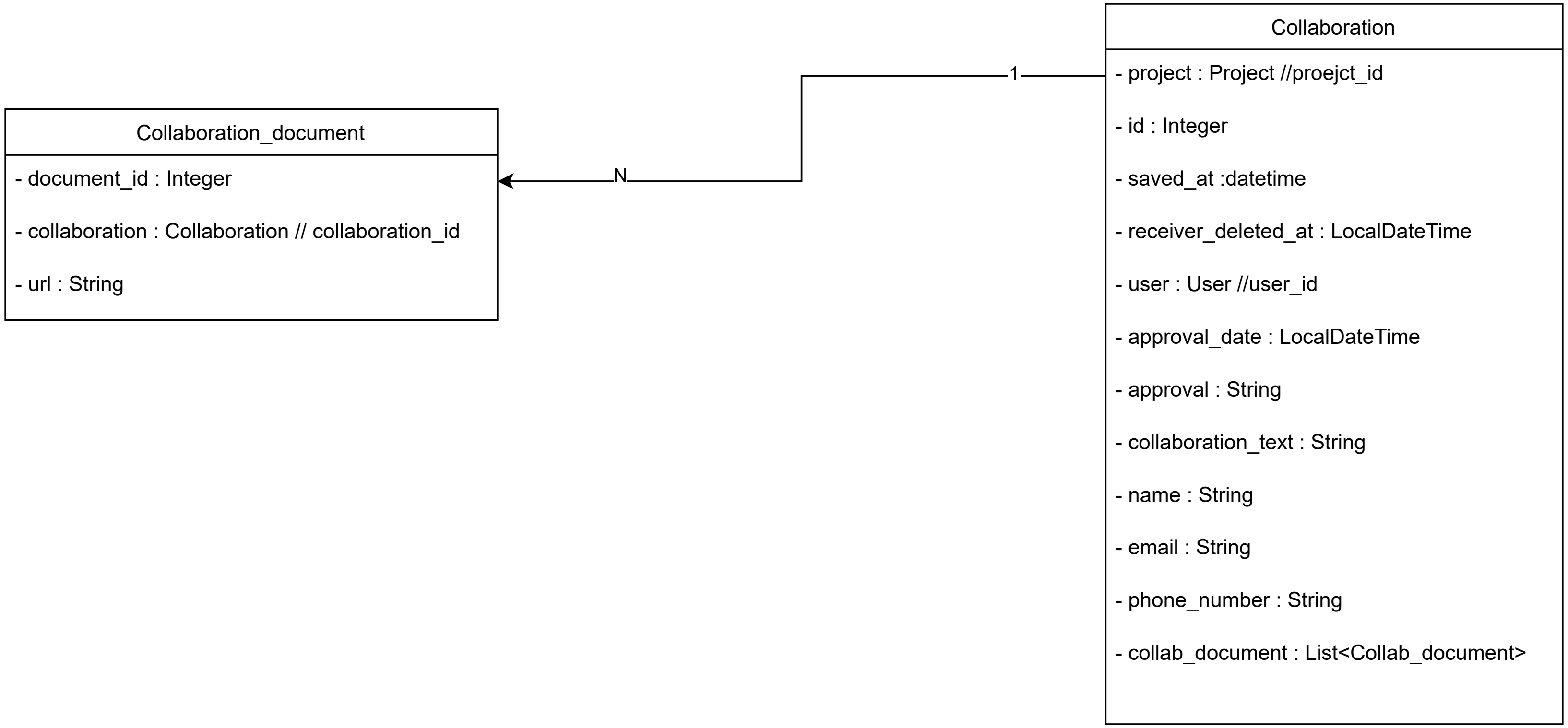
liked_project

- id: LikedProjectId

- member : Member //member_id

- project : Project //project_id

일[



Collaboration_document

- document_id : Integer

- collaboration : Collaboration // collaboration_id

- url : String

Collaboration

- project : Project //proejct_id

- id : Integer

- saved_at :datetime

- receiver_deleted_at : LocalDateTime

- user : User //user_id

- approval_date : LocalDateTime

- approval : String

- collaboration_text : String

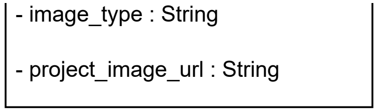
- name : String

- email : String

- phone_number : String

- collab_document : List<Collab_document>

개일



- image_type : String

- project_image_url : String