AuthenticationFacade

+ findById(id: UUID): Member

+ getIdentity(member: Member): Identity

+ login(username: String, password: String):
MemberWithToken

- findOrElseCreateMember(existingMember:
ExistingMemberEntity): MemberEntity

- createToken(memberId: UUID): String

AuthorizationFacade

+ hasAccess(access: Access, accesses:
List[Access]): Boolean

+ permissions(memberId: UUID, departmentId:
UUID): List[Permission]

+ accesses(memberId: UUID, departmentId:
UUID): List[Access]

+ accesses(memberId: UUID, departmentName:
String): List[Access]

- getAccessValidator(access:
Access)(memberAccesses: List[Access]):
Boolean

DepartmentFacade

+ find(id: UUID): Department

+ findAll: List[Department]

+ findByName(name: String): Department

+ roles(id: UUID): List[Role]

+ materials(id: UUID): List[Material]

+ spaces(id: UUID): List[Space]

+ create(input: CreateDepartmentInput):
Department

- isDepartmentNameValid(name: String):
Boolean

+ executeQuery(
Future[Json]+ f

+ executeQuery(
operationName:O
Option[Json]):
List[Role]

AuthenticationPersist

+ findByMemberId(memberId: UUID):
List[MemberDepartmentRelationEntity]

DepartmentPersist

+ find(id: UUID): Option[DepartmentEntity]

+ findAll: List[DepartmentEntity]

+ findByName(name: String):
Option[DepartmentEntity]

+ findByMemberId(id: UUID):
List[DepartmentEntity]

+ insert(departmentEntity: DepartmentEntity):
Boolean

MaterialPersist

+ find(id: UUID): Option[MaterialEntity]

+ findByDepartmentId(departmentId: UUID):
List[MaterialEntity]

+ create(materialEntity: MaterialEntity):
Boolean

+ delete(id: UUID): Boolean

+ find(id: UUID

+ findByUsernam
Option[MemberEn

+ findByRoleId(
List[MemberEnti

+ insert(member
Option[MemberEn

BaseFacade

+ findById(id: UUID): Member

+ getIdentity(member: Member): Identi

+ login(username: String, password: S

- findOrElseCreateMember(existingMemb
MemberEntity

- createToken(memberId: UUID): String

GraphqlFacade

Query(form: GraphqlQuery):
on]+ findAll: List[Department]

Query(document: Document,
Name:Option[String],variables:
on]): Future[Json]+ roles(id: UUID):
]

MaterialFacade

+ find(id: UUID):Material

+ create(input: CreateMaterialInput):Material

+ delete(input:
DeleteMaterialInput):Department

MemberFacade

+ find(id: UUID):Member

+ findByUsername(username: String):Member

+ requests(id: UUID):List[Request]

+ reviews(id: UUID):List[Review]

+ reservations(id: UUID):List[Reservation]

+ roles(id: UUID):List[Role]

+ departments(id: UUID):List[Department]

ProblemFacad

+ find(id: UUID): Try[Problem

+ create(input: CreateProblem
Try[Problem]

MemberPersist

: UUID): Option[MemberEntity]

Username(username: String):
mberEntity]

oleId(roleId: UUID):
erEntity]

member: MemberEntity):
mberEntity]

MemberRolePersist

+ find(memberId: UUID, roleId: UUID):
Option[MemberRoleEntity]

+ insert(memberRoleEntity: MemberRoleEntity):
Boolean

+ delete(memberId: UUID, roleId: UUID):
Boolean

ProblemPersist

+ find(id: UUID): Option[ProblemEntity]

+ findBySpaceId(spaceId: UUID):
List[ProblemEntity]

+ create(problem: ProblemEntity): Boolean

RequestPersis

+ find(id: UUID): Option[Memb

+ findByUsername(username: St
Option[MemberEntity]

+ findByRoleId(roleId: UUID):
List[MemberEntity]

+ insert(member: MemberEntity
Option[MemberEntity]

```
┌──────────────────────────────────────┐
│ Facade                               │
├──────────────────────────────────────┤
│                                      │
├──────────────────────────────────────┤
│  Identity                            │
│ ord: String): MemberWithToken        │
│ ngMember: ExistingMemberEntity):     │
│ String                               │
└──────────────────────────────────────┘
```

```
┌──────────────────┐   ┌──────────────────────────────────┐   ┌──────────────────────────────────────────────┐   ┌──────────────────────────────────────┐
│ mFacade          │   │          PermissionFacade         │   │                RequestFacade                  │   │            ReservationFacade           │
├──────────────────┤   ├──────────────────────────────────┤   ├──────────────────────────────────────────────┤   ├──────────────────────────────────────┤
│                  │   │                                  │   │                                              │   │                                      │
├──────────────────┤   ├──────────────────────────────────┤   ├──────────────────────────────────────────────┤   │                                      │
│                  │   │                                  │   │ + find(id: UUID):Request                     │   │                                      │
│ Problem]         │   │                                  │   │ + reviews(id: UUID):List[Review]             │   ├──────────────────────────────────────┤
│ ProblemInput):   │   │ + find(name: String):Permission  │   │ + create(input: CreateRequestInput):Request  │   │                                      │
│                  │   │ + permissions: List[Permission]  │   │ + cancel(input: CancelRequestInput):Member   │   │ + find(id: UUID):Reservation         │
│                  │   │                                  │   │ - isMemberOwnRequest(member: Member, request:│   │                                      │
│                  │   │                                  │   │ RequestEntity)                               │   │                                      │
└──────────────────┘   └──────────────────────────────────┘   └──────────────────────────────────────────────┘   └──────────────────────────────────────┘
```

```
┌──────────────────┐   ┌──────────────────────────────────┐   ┌──────────────────────────────────────────────┐   ┌──────────────────────────────────────┐
│ Persist          │   │         ReservationPersist        │   │                ReviewPersist                  │   │               RolePersist              │
├──────────────────┤   ├──────────────────────────────────┤   ├──────────────────────────────────────────────┤   ├──────────────────────────────────────┤
│                  │   │                                  │   │                                              │   │                                      │
├──────────────────┤   ├──────────────────────────────────┤   ├──────────────────────────────────────────────┤   │ + find(id: UUID): Option[RoleEntity] │
│ n[MemberEntity]  │   │ + find(id: UUID): Option[Reserv  │   │ + find(id: UUID): Option[ReviewEntity]       │   │ + findByName(departmentName: String, name:│
│ ame: String):    │   │   ationEntity]                   │   │ + findByRequestId(requestId: UUID):          │   │ String): Option[RoleEntity]          │
│                  │   │ + findBySpaceId(spaceId: UUID):  │   │ List[ReviewEntity]                           │   │ + findByDepartmentId(departmentId: UUID):│
│ UUID):           │   │ List[ReservationEntity]          │   │                                              │   │ List[RoleEntity]                     │
│                  │   │ + findByClientId(clientId: UUID):│   │ + findByReviewerId(reviewerId: UUID):        │   │ + findByMemberId(memberId: UUID):    │
│ rEntity):        │   │ List[ReservationEntity]          │   │ List[ReviewEntity]                           │   │ List[RoleEntity]                     │
│                  │   │ + insert(reservationEntity:      │   │ + insert(reviewEntity: ReviewEntity): Boolean│   │ + insert(roleEntity: RoleEntity): Boolean│
│                  │   │ ReservationEntity): Boolean      │   │                                              │   │ + update(roleEntity: RoleEntity): Boolean│
│                  │   │                                  │   │                                              │   │ + delete(id: UUID): Boolean          │
└──────────────────┘   └──────────────────────────────────┘   └──────────────────────────────────────────────┘   └──────────────────────────────────────┘
```

RoleFacade

+ find(id: UUID): Try[Role]

+ findByName(department: String,
            name: String): Role

+ members(id: UUID): List[Member]

+ create(input: CreateRoleInput):Role

+ update(input: UpdateRoleInput):Role

+ delete(input: DeleteRoleInput):Department

+ assign(input: AssignRoleInput):Role

+ revoke(input: RevokeRoleInput):Role

ReviewFacade

+ find(id: UUID): Try[Review]

+ create(input: CreateReviewInput):Review

- updateRequestStatus(requestId: UUID):
Boolean

- updateReservation(requestId: UUID): Boolean

- findRequest(requestId: UUID):RequestEntity

- createReview(reviewEntity: ReviewEntity):
ReviewEntity

- createReservations(requestEntity:
RequestEntity): Boolean

- requiredApproval: Int

SpaceFacade

+ find(id: UUID): Try[Space]

+ findByName(department: String, name:
String):Space

+ searchByName(name: String):List[Space]

+ findAll: List[Space]

+ requests(space: Space): List[Request]

+ reservations(space: Space):
List[Reservation]

+ problems(space: Space):List[Problem]

+ create(input: CreateSpaceInput):Space

+ update(input: UpdateSpaceInput):Space

- isSpaceNameValid(name: String):Boolean

SearchPersist

+ space(query: String,
        department: Option[String],
        tags: List[String],
        capacity: Option[Int]):
List[SpaceEntity]

SearchPersist

+ find(id: UUID): Option[SpaceEntity]

+ searchByName(name: String):
List[SpaceEntity]

+ findByName(departmentName: String,
spaceName: String): Option[SpaceEntity]

+ findByDepartmentId(departmentId: UUID):
List[SpaceEntity]

+ findAll: List[SpaceEntity]

+ insert(space: SpaceEntity): Boolean

+ update(space: SpaceEntity): Boolean

+ delete(id: UUID): Boolean