

OASIS详细设计描述文档

文档修改历史

修改人员	日期	修改原因	版本号
田晨江	2020.3.3	创建文档	1.0
陈耿阳、田晨江、赵文祺、孙逸伦	2020.3.7	评审文档	1.1
田晨江	2020.4.1	根据迭代二补充文档	2.0
陈耿阳、田晨江、赵文祺、孙逸伦	2020.4.5	评审文档	2.2
田晨江	2020.6.27	根据迭代三补充文档	3.0

目录

- OASIS详细设计描述文档
 - 文档修改历史
 - 目录
 - 1. 引言
 - 1.1 编制目的
 - 1.2 词汇表
 - 1.3 参考资料
 - 2. 产品描述
 - 3. 系统结构设计概述
 - 4. 结构视角
 - 4.1 业务逻辑层的分解
 - 4.1.1 paperbl模块
 - 4.1.2 adminbl模块
 - 4.1.3 searchbl 模块
 - 4.1.4 rankbl 模块
 - 4.1.5 portraitbl 模块
 - 4.1.6 picturebl 模块
 - 4.1.7 conferencebl 模块
 - 4.1.8 taskbl 模块
 - 4.1.9 talentbl 模块
 - 5. 依赖视角

1. 引言

1.1 编制目的

本报告详细完成对OASIS系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注
OASIS	学术关系图谱系统	无

1.3 参考资料

(1) 《软件工程与计算(卷二)》

(1) 《软件工程与计算(卷三)》

2. 产品描述

参考OASIS需求规格说明文档中对产品的概括，描述。

3. 系统结构设计概述

参考OASIS系统体系结构文档中对于体系结构设计的概述。

4. 结构视角

4.1 业务逻辑层的分解

业务逻辑层的开发包图参见软件体系结构文档图2。

4.1.1 paperbl模块

(1) 模块概述

paperbl模块承担的需求见需求规格说明文档功能需求及相关非功能需求。

paperbl模块的职责及接口参见软件系统结构描述文档表15。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。restful api层和业务逻辑层之间添加 bl.paperBLSERVICE接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

paperbl模块的设计如图1所示：

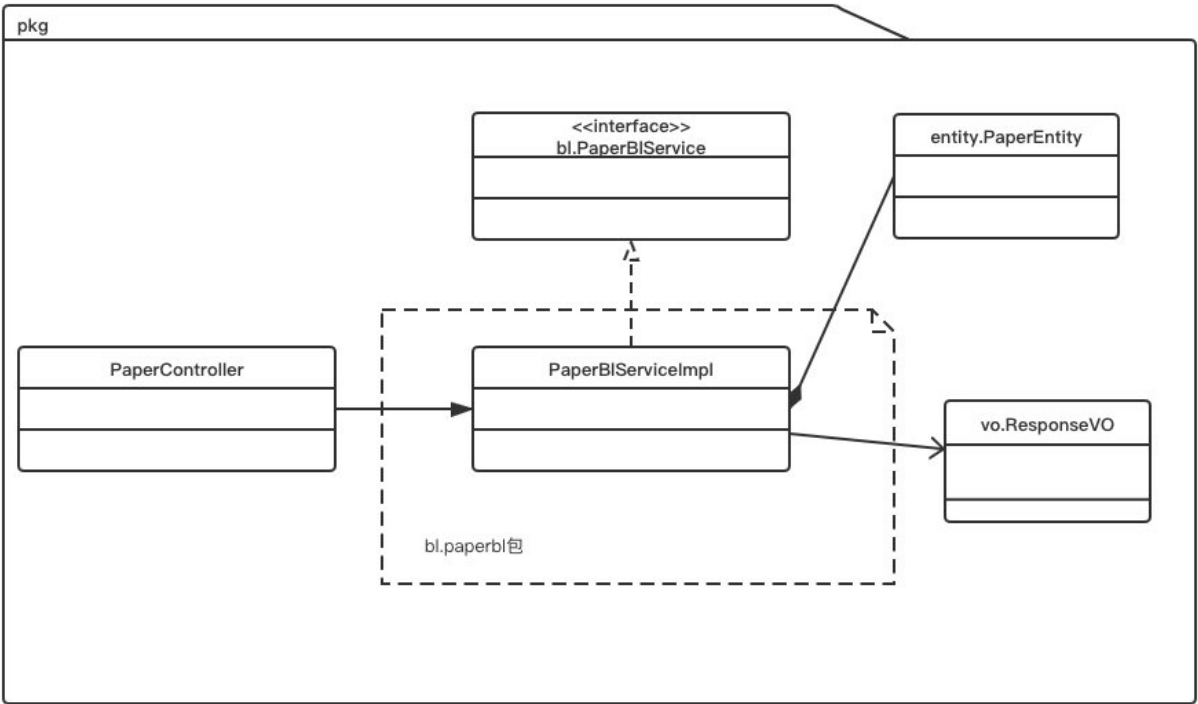


图1 paperbl模块各个类的设计

paperbl模块的各个类的职责如表1所示：

表1 paperbl模块各个类的职责

模块	职责
PaperBLSERVICEImpl	负责实现论文相关内容所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的相关信息

(3) 模块内部类的接口规范

PaperBLSERVICEImpl 的接口规范如表2所示：

表2 PaperBLSERVICEImpl的接口规范

提供的服务（供接口）		
		public BasicResponse

PaperBLServiceImpl.getResearcherInterest	语法	getResearcherInterest(String id);
前置条件	传入符合条件的作者id	
后置条件	无	
PaperBLServiceImpl.getAffiliationInterest	语法	public BasicResponse getAffiliationInterest(String id)
前置条件	传入符合条件的机构名	
后置条件	无	
PaperBLServiceImpl.getJournalInterest	语法	public BasicResponse getJournalInterest(String id)
前置条件	传入符合条件的期刊名	
后置条件	无	
PaperBLServiceImpl.getConferenceInterest	语法	public BasicResponse getConferenceInterest(String id)
前置条件	传入符合条件的会议名	
后置条件	无	
PaperBLServiceImpl.getActivePaperAbstract	语法	public BasicResponse getActivePaperAbstract();
前置条件	无	
后置条件	无	
PaperBLServiceImpl.getReferenceById	语法	public BasicResponse getReferenceById(String paperId);
前置条件	传入符合条件的论文id	
后置条件	无	
PaperBLServiceImpl.getAuthorPapersById	语法	public BasicResponse getAuthorPapersById(String authorId, int page, String sortKey);
前置条件	传入符合条件的作者 id, 页数, 规定条件的 sortkey	
后置条件	无	
PaperBLServiceImpl.getAffiliationPapers	语法	public BasicResponse getAffiliationPapers(String affiliation, int page, String sortKey);

前置条件	传入符合条件的机构id, 页数, 规定条件的sortkey	
后置条件	无	
PaperBLServicImpl.getKeywordPapers	语法	public BasicResponse getKeywordPapers(String keyword, int page, String sortKey);
前置条件	无	
后置条件	无	
PaperBLServicImpl.getSimilarAuthors	语法	public BasicResponse getSimilarAuthors(String authorId)
前置条件	无	
后置条件	无	
PaperBLServicImpl.getKeyword3DTrend()	语法	public BasicResponse getKeyword3DTrend()
前置条件	无	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.aggregate()	根据聚合条件对数据集进行聚合操作	
mongoTemplate.find()	查找符合条件的结果集	
mongoTemplate.findOne()	查找符合条件的结果集的第一条	
mongoTemplate.count()	计算符合条件的结果集大小	

(4) 业务逻辑层的动态模型

图2表明了OASIS系统中, 当发出查找某个学者研究方向时, 论文业务逻辑处理的相关对象之间的协作。

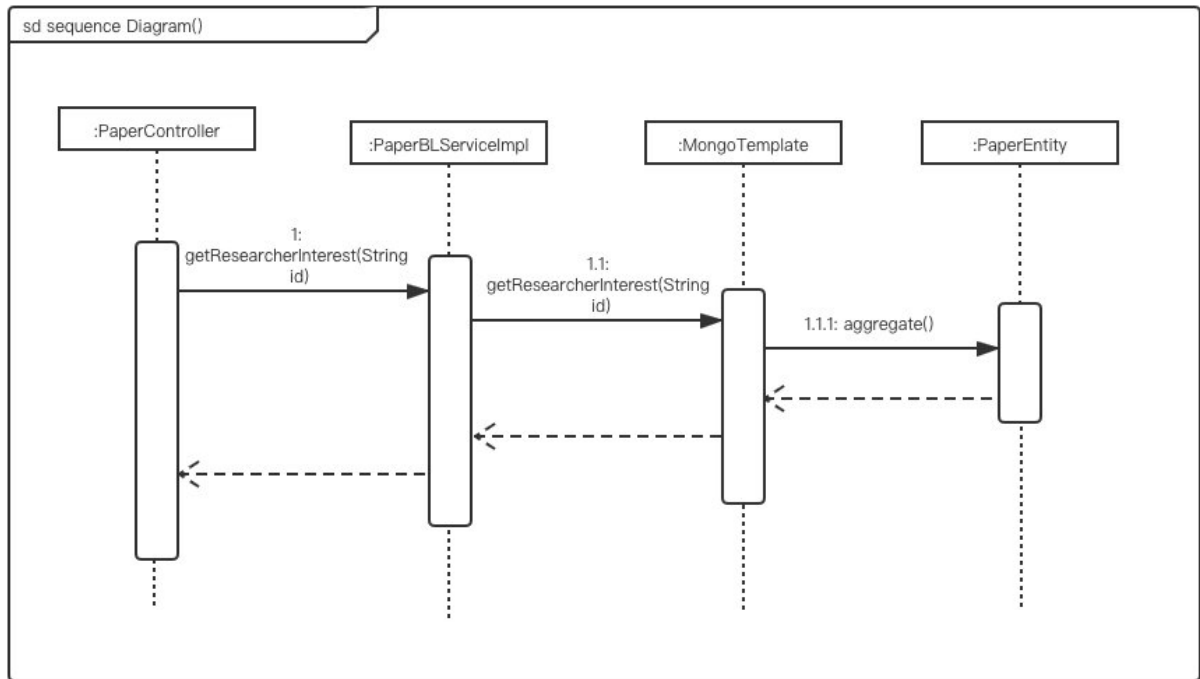


图2 得到学者研究方向的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的BLServicImpl委托给不同的领域对象。

4.1.2 adminbl模块

(1) 模块概述

adminbl模块承担的需求见需求规格说明文档功能需求及相关非功能需求。

adminbl模块的职责及接口参见软件系统结构描述文档表16。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。restful api层和业务逻辑层之间添加bl.AdminBLService接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

adminbl模块的设计如图3所示：

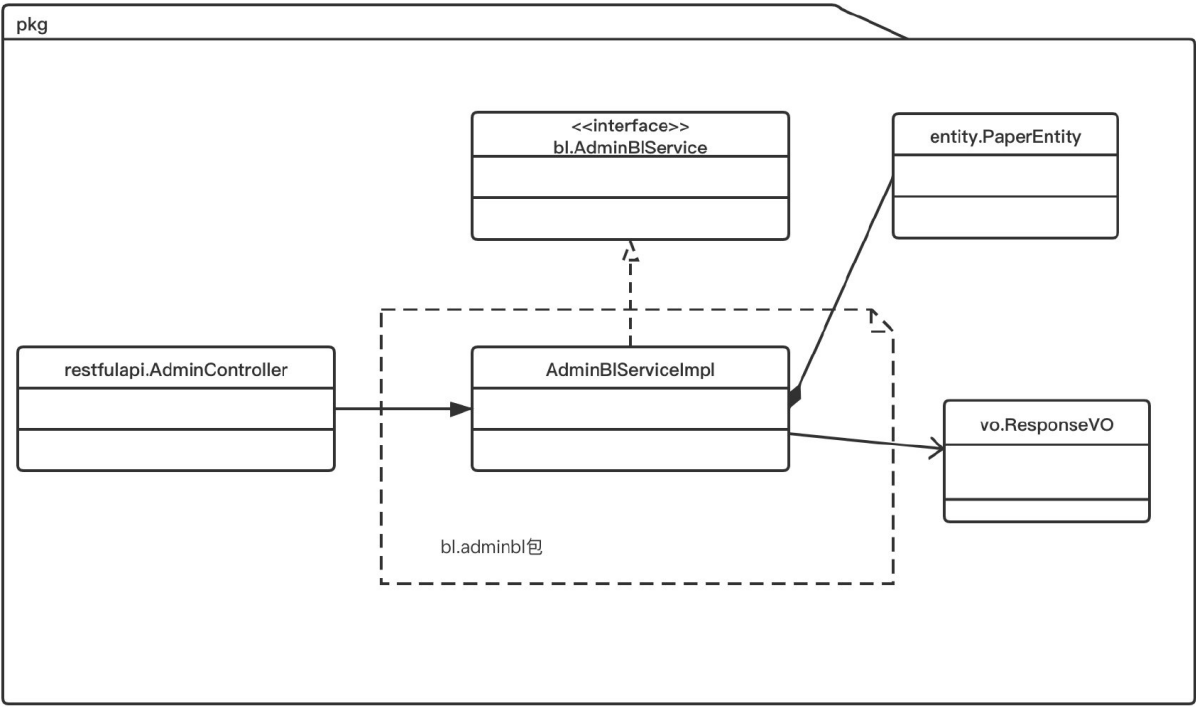


图3 adminbl模块各个类的设计

adminbl模块各个类的职责如表3所示：

表3 adminbl模块各个类的职责

职责	模块
AdminBIServiceImpl	负责实现管理员相关所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的身份信息

(3) 模块内部类的接口规范

AdminBIServiceImpl 的接口规范如表4所示：

表4 AdminBIServiceImpl的接口规范

提供的服务（供接口）		
AdminBIServiceImpl.importPaperData	语法	public BasicResponse importPaperData(MultipartFile file);
前置条件	传入符合要求的file	
后置条件	导入数据	
AdminBIServiceImpl.getConferenceInfo	语法	public BasicResponse getConferenceInfo(int page, String name);
前置条件	传入符合条件的会议名和页数	

后置条件	无	
AdminBLServicImpl.getAffiliationInfo	语法	public BasicResponse getAffiliationInfo(int page, String name);
前置条件	传入符合条件的机构名和页数	
后置条件	无	
AdminBLServicImpl.getJournalInfo	语法	public BasicResponse getJournalInfo(int page, String name);
前置条件	传入符合条件的页数和期刊名	
后置条件	无	
AdminBLServicImpl.getAuthorInfo	语法	public BasicResponse getAuthorInfo(int page, String name);
前置条件	传入符合条件的页数和作者名	
后置条件	无	
AdminBLServicImpl.mergeAffiliationInfo	语法	public BasicResponse mergeAffiliationInfo(List src, String desc);
前置条件	传入符合条件的待修改名字和要修改的名字	
后置条件	无	
AdminBLServicImpl.updateConferenceInfo	语法	public BasicResponse updateConferenceInfo(String src, String desc);
前置条件	传入符合条件的要修改的会议名	
后置条件	无	
AdminBLServicImpl.updateJournalInfo	语法	public BasicResponse updateJournalInfo(String src, String desc);
前置条件	传入符合条件的要修改的期刊名	
后置条件	无	
AdminBLServicImpl.updatePaperInfo	语法	public BasicResponse updatePaperInfo(UpdatePaperParameter parameter);
前置条件	无	
后置条件	无	
AdminBLServicImpl.mergeAuthorInfo	语法	public BasicResponse mergeAuthorInfo(List src, String desc);
前置条件	传入符合条件的要修改的作者名和要修改的名字	

后置条件	无	
AdminBLSERVICEImpl.getKeywordInfo	语法	public BasicResponse getKeywordInfo(int page, String name)
前置条件	传入符合条件的关键词和页数	
后置条件	无	
AdminBLSERVICEImpl.getRecommendedSimilarAffiliation	语法	public BasicResponse getRecommendedSimilarAffiliation()
前置条件	无	
后置条件	无	
AdminBLSERVICEImpl.getRecommendedSimilarAuthor	语法	public BasicResponse getRecommendedSimilarAuthor()
前置条件	无	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.aggregate()	根据聚合条件对数据集进行聚合操作	
mongoTemplate.find()	查找符合条件的结果集	
mongoTemplate.findOne()	查找符合条件的结果集的第一条	
mongoTemplate.count()	计算符合条件的结果集大小	

(4) 业务逻辑层的动态模型

图4表明了OASIS系统中，当管理员导入论文数据之后，论文业务逻辑处理的相关对象之间的协作。

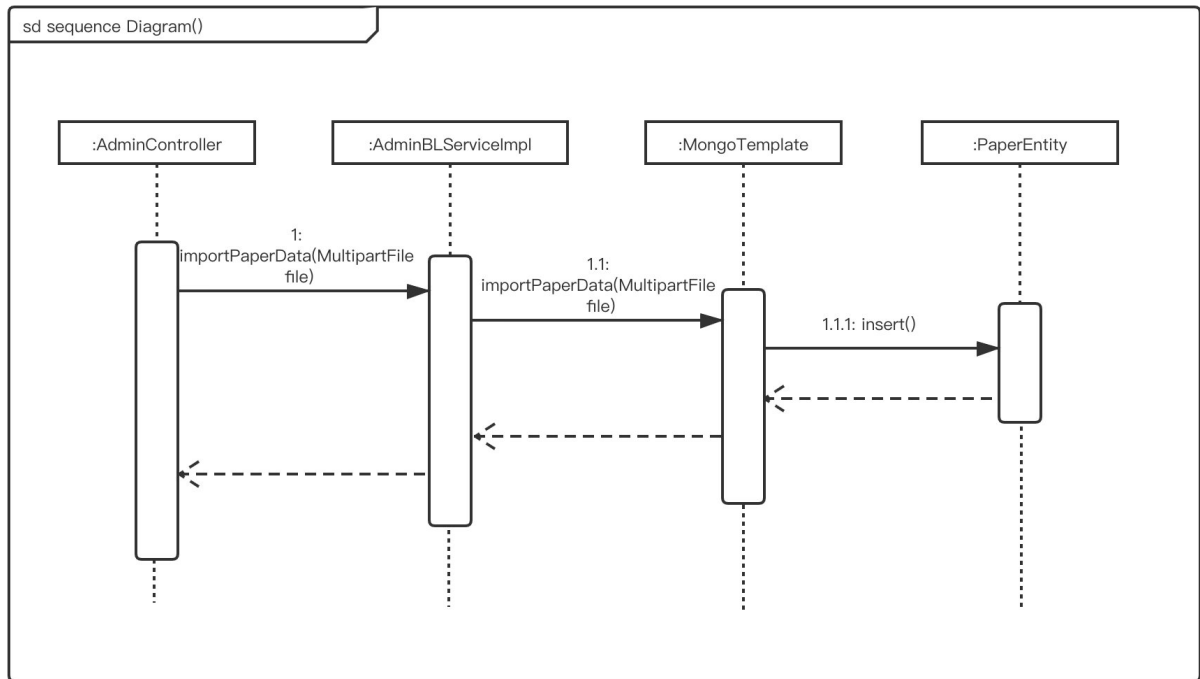


图4 导入论文的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的BLServicImpl委托给不同的领域对象。

4.1.3 searchbl 模块

(1) 模块概述

search 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

search 模块的职责及接口参见软件系统结构描述文档表17。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.SearchBlService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

search 模块的设计如图5所示：

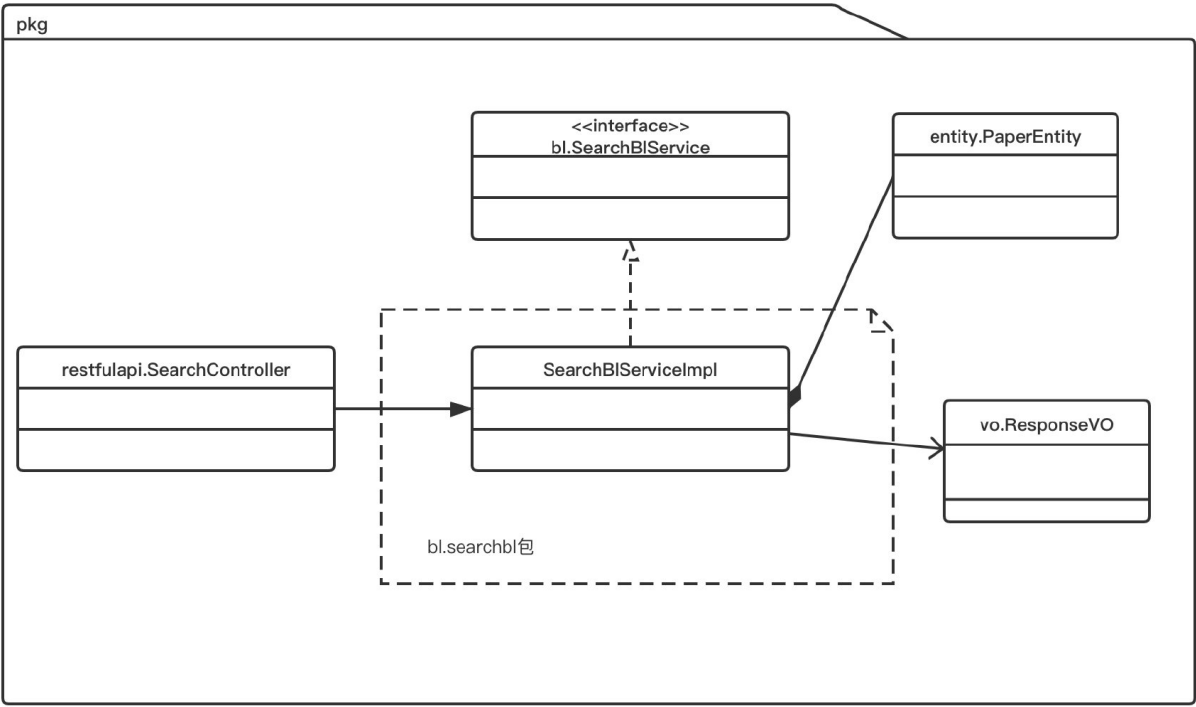


图5 searchbl模块各个类的设计

searchbl模块各个类的职责如表5所示：

表5 searchbl模块各个类的职责

模块	职责
SearchBIServiceImpl	负责实现检索论文所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的相关信息

(3) 模块内部类的接口规范

SearchServiceImpl 的接口规范如表6所示：

表6 SearchServiceImpl 的接口规范

提供的服务（供接口）		
SearchBLSERVICEImpl.basicSearchByES	语法	public BasicResponse BasicResponse basicSearchByES(String keyword, int page, String sortKey);
前置条件	传入符合条 件的页数和 排序标准	
后置条件	无	
SearchBLSERVICEImpl.advancedSearchByES	语法	public BasicResponse advancedSearchByES(String author, String affiliation, String publicationName, String keyword, int startYear, int endYear, int page, String sortKey);
前置条件	传入符合条 件的页数和 排序标准	
后置条件	无	
SearchBLSERVICEImpl.getBasicSearchFilterCondition	语法	public BasicResponse getBasicSearchFilterCondition(String keyword);
前置条件	无	
后置条件	无	
SearchBLSERVICE.commandSearch(String query, int page, String sortKey)	语法	public BasicResponse commandSearch(String query, int page, String sortKey)
前置条件	传入符合条 件的查询语 句，页数和 排序标准	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.find()	查找满足查 询条件的数 据	
mongoTemplate.count()	统计满足条 件的结果条 数	
RestHighLevelClient.search()	通过ES进行 搜索	

(4) 业务逻辑层的动态模型

图6表明了OASIS系统中，当用户做了基础查询之后，查询业务逻辑处理的相关对象之间的协作。

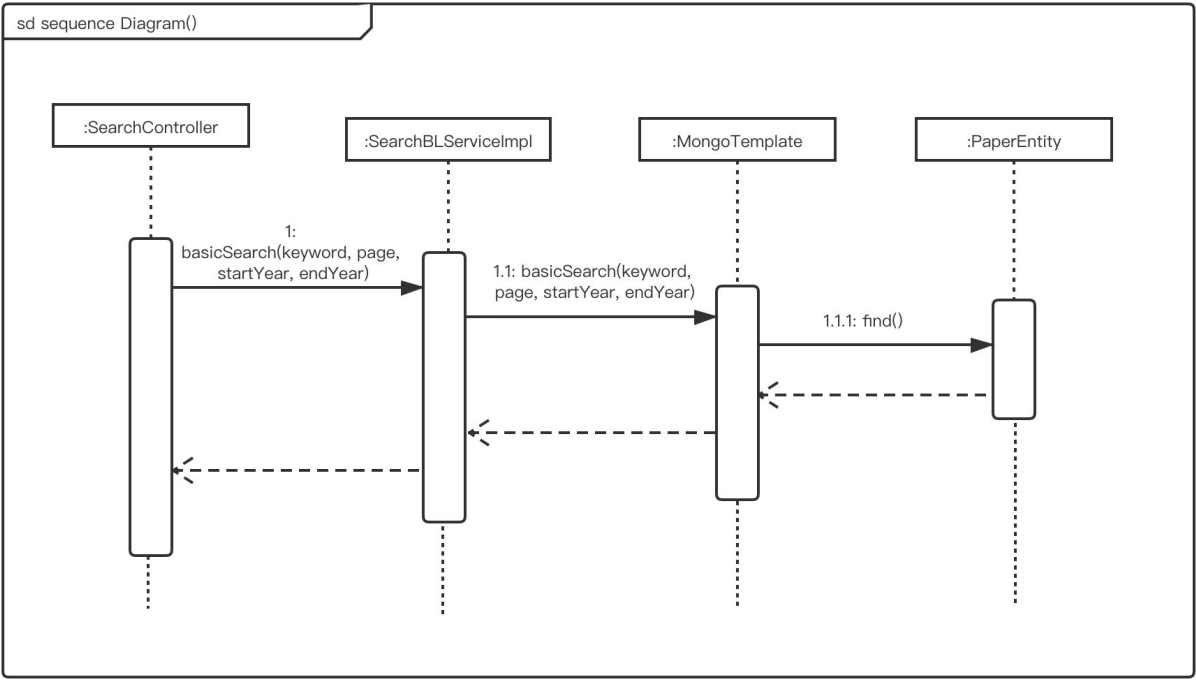


图6 基础查询的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的BLServletImpl委托给不同的领域对象。

4.1.4 rankbl 模块

(1) 模块概述

rank 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

rank 模块的职责及接口参见软件系统结构描述文档表18。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.RankService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

rank 模块的设计如图7所示

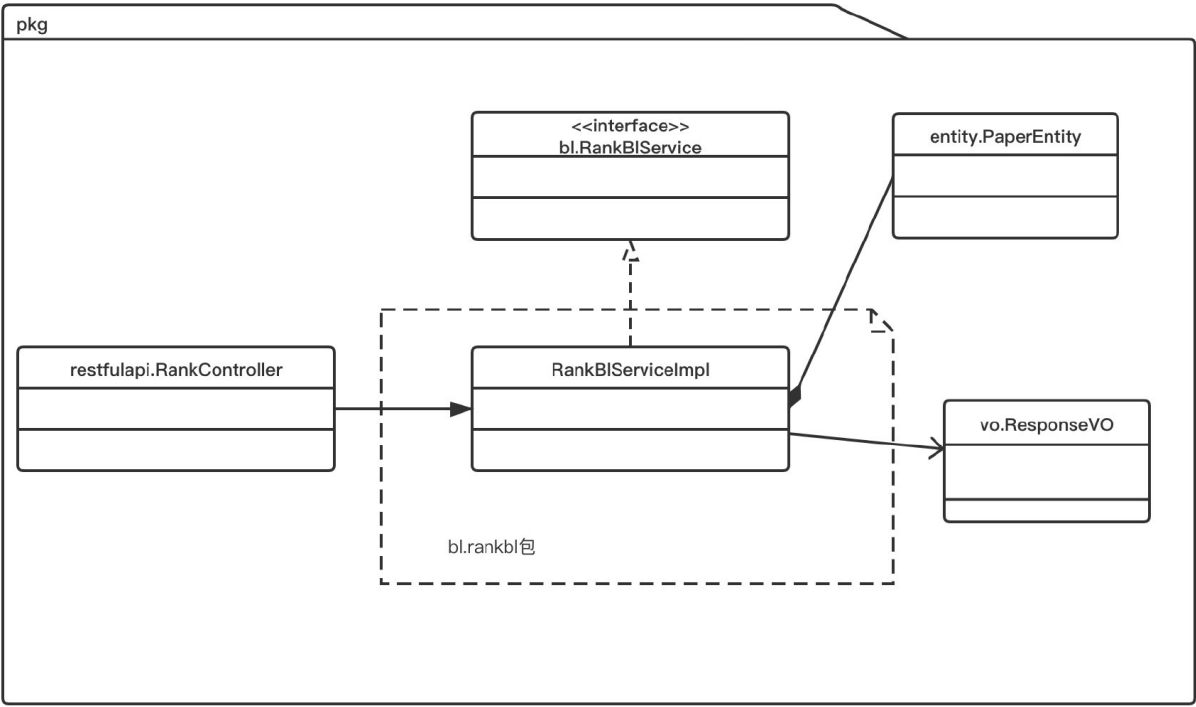


图7 rankbl模块各个类的设计

rankbl模块各个类的职责如表7所示：

表7 rankbl模块各个类的职责

模块	职责
RankBLSERVICEImpl	负责实现排名信息所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的相关信息

(3) 模块内部类的接口规范

RankBLSERVICEImpl 的接口规范如表8所示：

表8 RankBLSERVICEImpl模块的接口规范

提供的服务（供接口）		
RankBLSERVICEImpl.getAffiliationBasicRanking	语法	public BasicResponse getAffiliationBasicRanking(String sortKey, String year);
前置条件	传入符合条件的排序标准和年份	
后置条件	无	
RankBLSERVICE.getAuthorBasicRanking	语法	public BasicResponse getAuthorBasicRanking(String sortKey, String year);

前置条件	传入符合条件的排序标准和年份	
后置条件	无	
RankBLServiceImpl.getJournalBasicRanking	语法	public BasicResponse getJournalBasicRanking(String sortKey, int year);
前置条件	传入符合条件的排序标准和年份	
后置条件	无	
RankBLServiceImpl.getConferenceBasicRanking	语法	public BasicResponse getConferenceBasicRanking(String sortKey, int year);
前置条件	传入符合条件的排序标准和年份	
后置条件	无	
RankBLServiceImpl.getKeywordBasicRanking	语法	public BasicResponse getKeywordBasicRanking(int year);
前置条件	传入符合条件的年份	
后置条件	无	
RankBLServiceImpl.getAuthorAdvancedRanking	语法	public BasicResponse getAuthorAdvancedRanking(String sortKey, int startYear, int endYear);
前置条件	传入符合条件的排序标准和起止年份	
后置条件	无	
RankBLServiceImpl.getAffiliationAdvancedRanking	语法	public BasicResponse getAffiliationAdvancedRanking(String sortKey, int startYear, int endYear);
前置条件	传入符合条件的排序标准和起止年份	
后置条件	无	
RankBLServiceImpl.getAffiliationDetailRankingById	语法	public BasicResponse getAffiliationDetailRankingById(String id);
前置条件	传入符合条件的机构名	
后置条件	无	
RankBLServiceImpl.getAuthorDetailRanking	语法	public BasicResponse getAuthorDetailRanking(String affiliation);
前置条件	传入符合条件的作者id	
后置条件	无	
RankBLServiceImpl.getAuthorDetailRankingById	语法	public BasicResponse getConferencePortrait(String conference)

前置条件	传入符合条件的作者id	
后置条件	无	
RankBLService.getKeywordAdvancedRanking	语法	public BasicResponse getKeywordAdvancedRanking(String sortKey, int startYear, int endYear)
前置条件	传入符合条件的排序标准和起止年份	
后置条件	无	
RankBLService.getAuthorDetailRankingByKeyword	语法	public BasicResponse getAuthorDetailRankingByKeyword(String keyword, String sortKey, int startYear, int endYear)
前置条件	传入符合条件的排序标准和起止年份	
后置条件	无	
RankBLService.getAffiliationDetailRankingByKeyword	语法	public BasicResponse getAffiliationDetailRankingByKeyword(String keyword, String sortKey, int startYear, int endYear)
前置条件	传入符合条件的排序标准和起止年份	
后置条件	无	
RankBLService.getKeywordDetailRankingById	语法	public BasicResponse getKeywordDetailRankingById(String keyword)
前置条件	无	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.aggregate()	根据聚合条件对数据集进行聚合操作	
mongoTemplate.find()	查找符合条件的结果集	
mongoTemplate.findOne()	查找符合条件的结果集的第一条	
mongoTemplate.count()	计算符合条件的结果集大小	

(4) 业务逻辑层的动态模型

图8表明了OASIS系统中，当用户发出得到作者排名的请求后，排名信息相关业务逻辑处理的相关对象之间的协作。

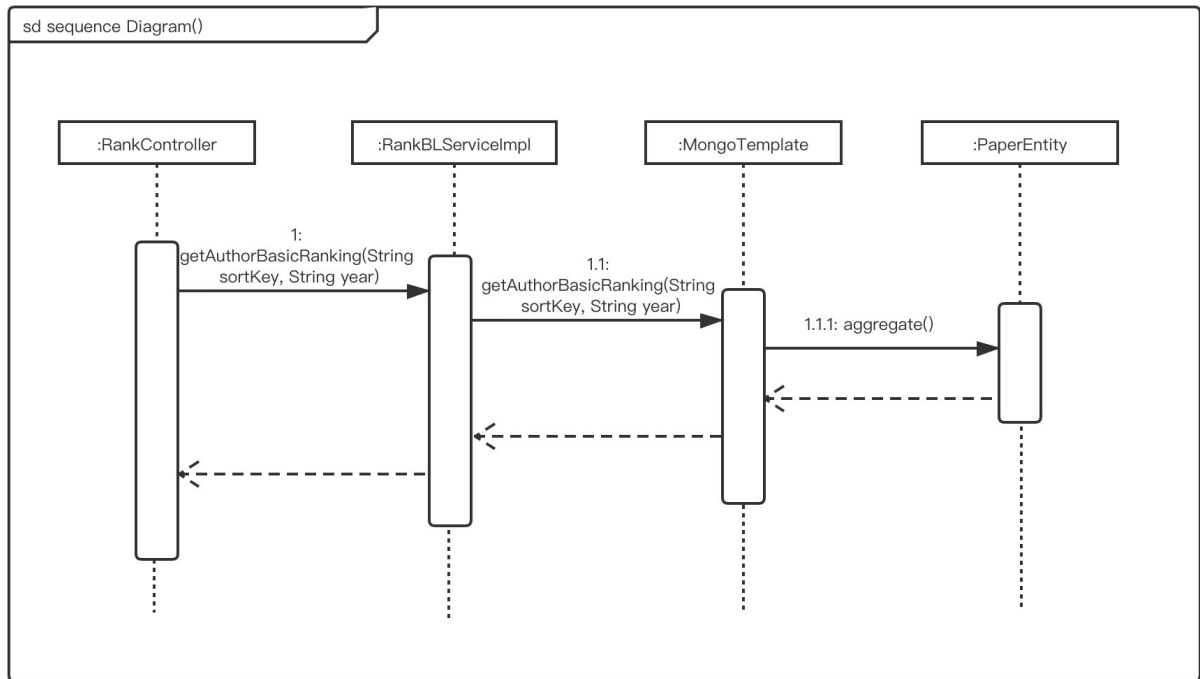


图8 获得作者排名的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

4.1.5 portraitbl 模块

(1) 模块概述

portrait 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

portrait 模块的职责及接口参见软件系统结构描述文档表19。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.PortraitService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

portrait 模块的设计如图16所示

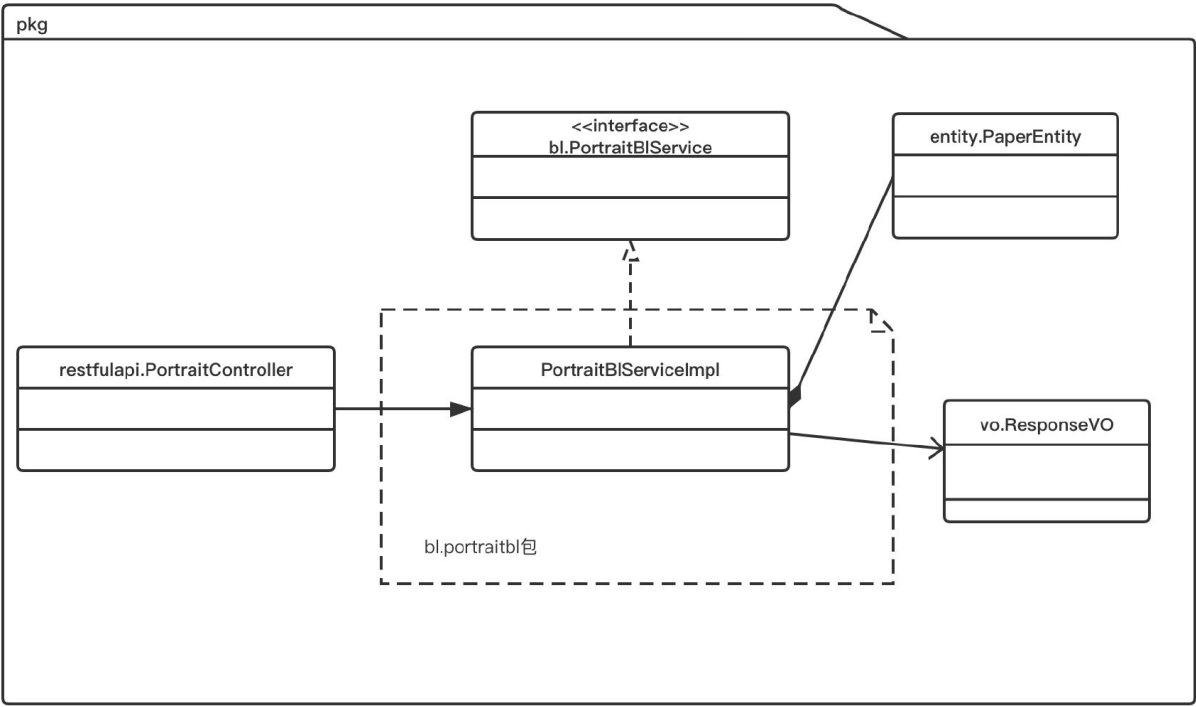


图9 portraitbl模块各个类的设计

portraitbl模块各个类的职责如表9所示：

表9 potraitbl模块各个类的职责

模块	职责
PortraitBIServiceImpl	负责实现排名信息所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的相关信息

(3) 模块内部类的接口规范

PortraitBIServiceImpl 的接口规范如表10所示：

表10 PortraitBIServiceImpl模块的接口规范

提供的服务（供接口）		
PortraitBIServiceImpl.getAuthorPortraitById	语法	public BasicResponse getAuthorPortraitById(String id);
前置条件	传入符合条件的作者id	
后置条件	无	
		public BasicResponse

PortraitBLServiceImpl.getAffiliationPortrait	语法	getAffiliationPortrait(String affiliation);
前置条件	传入符合条件的机构名	
后置条件	无	
PortraitBLServiceImpl.getKeywordPortrait	语法	public BasicResponse getKeywordPortrait(String keyword);
前置条件	传入符合条件的关键词	
后置条件	无	
PortraitBLServiceImpl.getConferencePortrait	语法	public BasicResponse getConferencePortrait(String conference);
前置条件	传入符合条件的会议名	
后置条件	无	
RankBLServiceImpl.getJournalPortrait	语法	public BasicResponse getJournalPortrait(String journal)
前置条件	传入符合条件的期刊名	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.aggregate()	根据聚合条件对数据集进行聚合操作	
mongoTemplate.find()	查找符合条件的结果集	
mongoTemplate.findOne()	查找符合条件的结果集的第一条	
mongoTemplate.count()	计算符合条件的结果集大小	

(4) 业务逻辑层的动态模型

图8表明了OASIS系统中，当用户发出查看某个作者的用户画像请求后，用户画像相关业务逻辑处理的相关对象之间的协作。

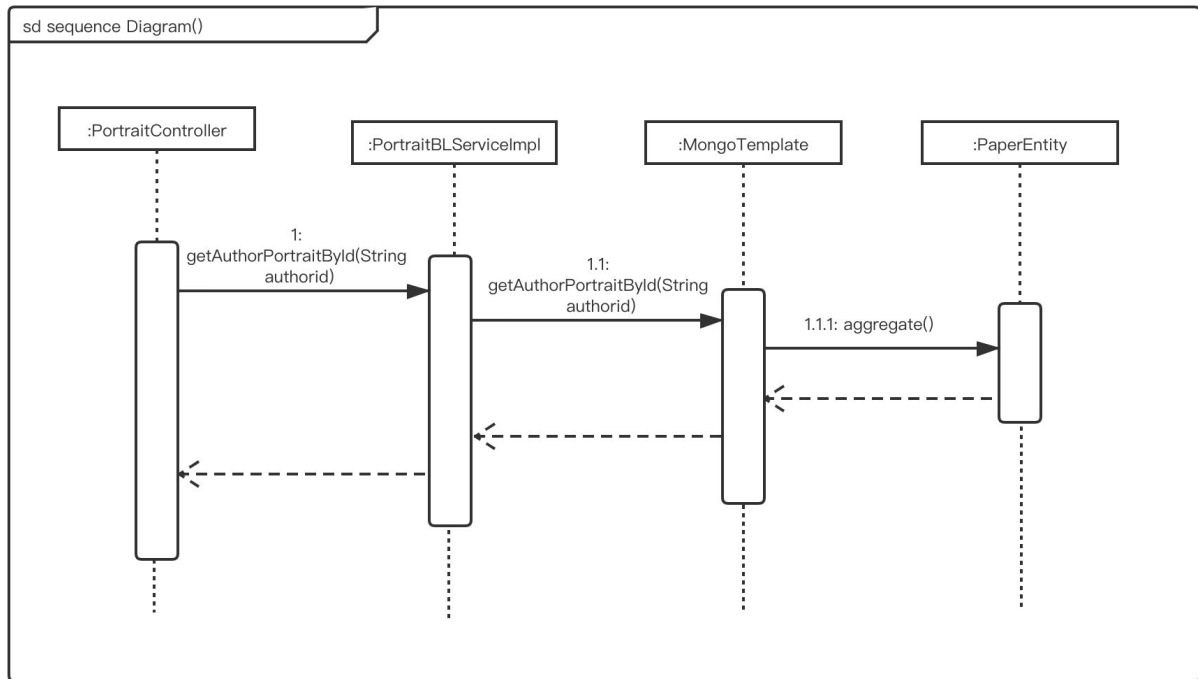


图10 获得作者用户画像的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

4.1.6 picturebl 模块

(1) 模块概述

picture 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

picture 模块的职责及接口参见软件系统结构描述文档表20。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.PictureService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

picture 模块的设计如图7所示

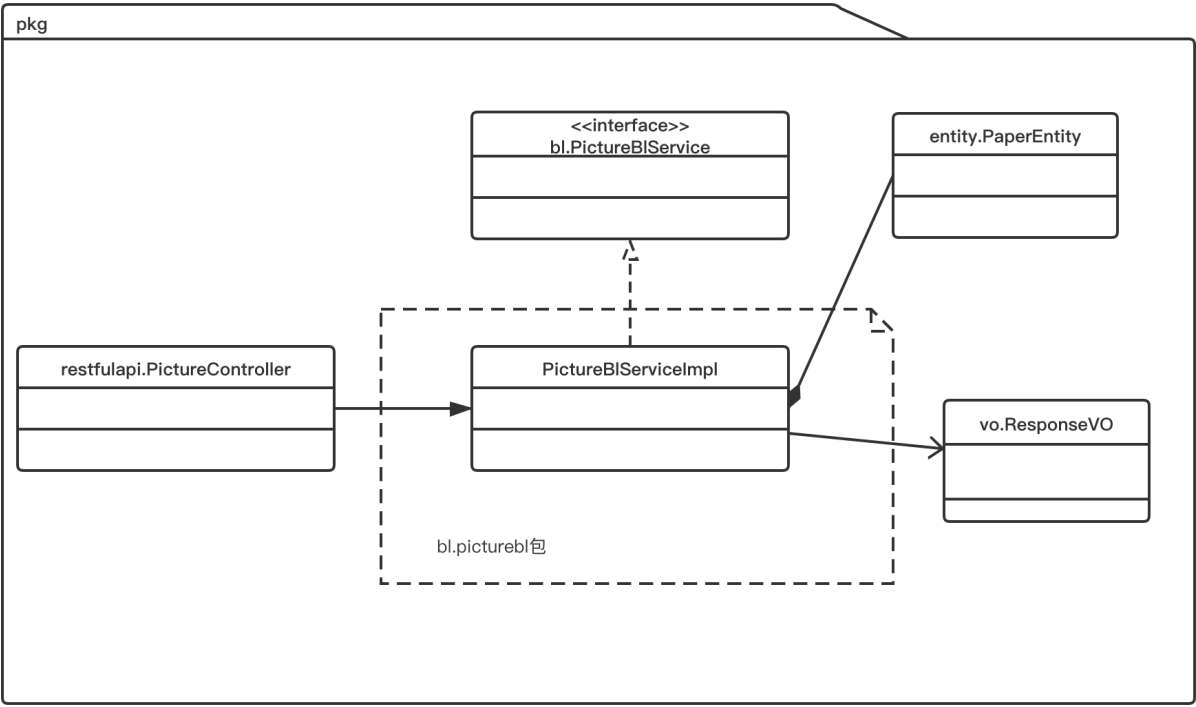


图11 picturebl模块各个类的设计

picturebl模块各个类的职责如表11所示：

表11 picturebl模块各个类的职责

模块	职责
PictureBLServiceImpl	负责实现排名信息所需要的服务
PaperEntity	论文的领域模型对象，拥有论文的相关信息

(3) 模块内部类的接口规范

PictureBLServiceImpl 的接口规范如表12所示：

表12 PictureBLServiceImpl模块的接口规范

提供的服务（供接口）		
PictureBLServiceImpl.getAcademicRelationByAuthorId	语法	public BasicResponse getAcademicRelationByAuthorId(String id);
前置条件	传入符合条件的作者 id	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.find()	查找符合条件的结果 集	
mongoTemplate.count()	计算符合条件的结果 集大小	

(4) 业务逻辑层的动态模型

图12表明了OASIS系统中，当用户发出查看某个作者的学术关系图谱请求后，图谱相关业务逻辑处理的相关对象之间的协作。

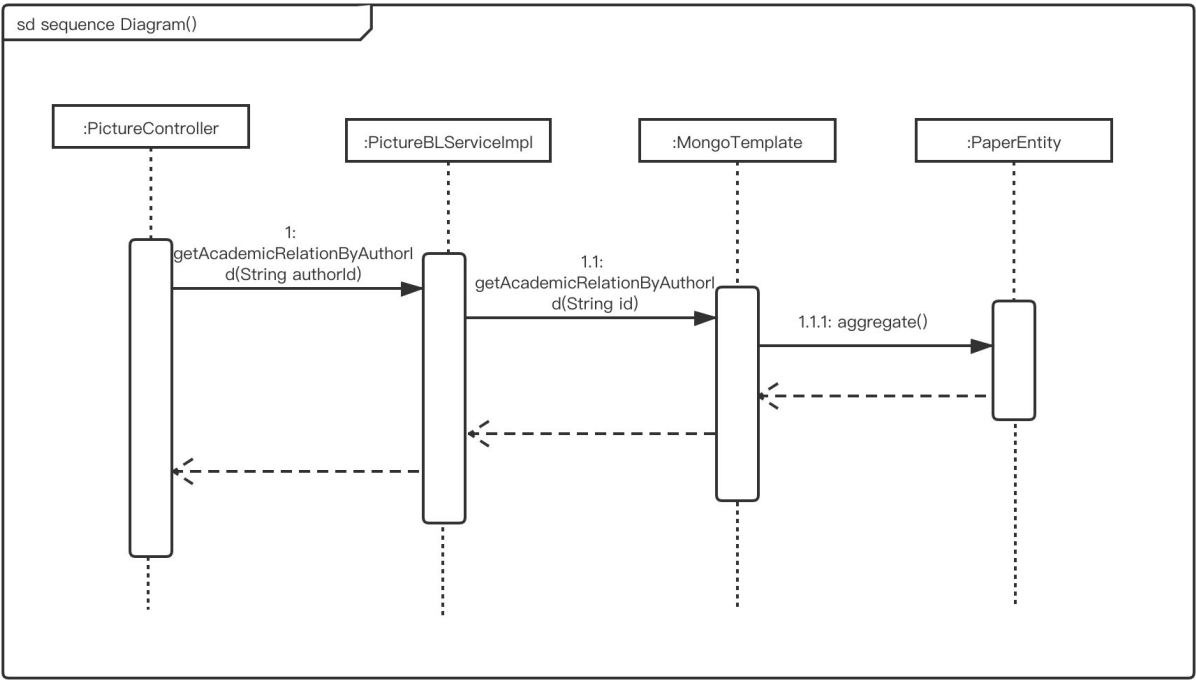


图12 获得作者排名的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

4.1.7 conferencebl 模块

(1) 模块概述

conference 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

conference 模块的职责及接口参见软件系统结构描述文档表21。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.ConferenceService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

conference 模块的设计如图7所示

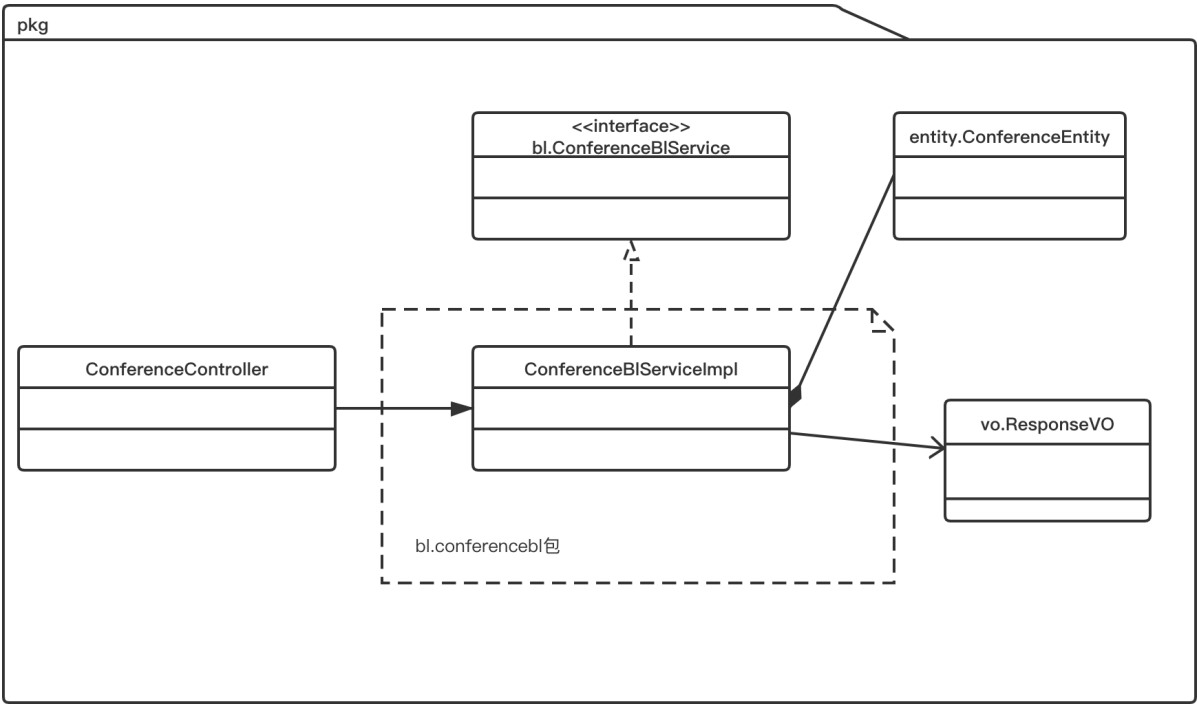


图13 conferencebl模块各个类的设计

conferencebl模块各个类的职责如表11所示：

表13 conferencebl模块各个类的职责

模块	职责
ConferenceBLSERVICEImpl	负责实现获取会议相关需要的服务
ConferenceEntity	会议的领域模型对象，拥有回忆的相关信息

(3) 模块内部类的接口规范

ConferenceBLSERVICEImpl 的接口规范如表12所示：

表14 ConferenceBLServiceImpl模块的接口规范

表21 ConferenceBLService模块的接口规范		
提供的服务（供接口）		
ConferenceBLServiceImpl.getConferencesAndJournalsList	语法	public BasicResponse getConferencesAndJournalsList(String keyword, int page)
前置条件	传入符合条件的页 数	
后置条件	无	
ConferenceBLServiceImpl.getConferencesAndJournalsProceedings	语法	public BasicResponse getConferencesAndJournalsProceedings(String titleId)
前置条件	传入符合条件的 title Id	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.find()	查找符合条件的结 果集	
mongoTemplate.count()	计算符合条件的结 果集大小	

(4) 业务逻辑层的动态模型

图12表明了OASIS系统中，当管理员发出查看包含某个关键词的会议或期刊列表请求后，图谱相关业务逻辑处理的相关对象之间的协作。

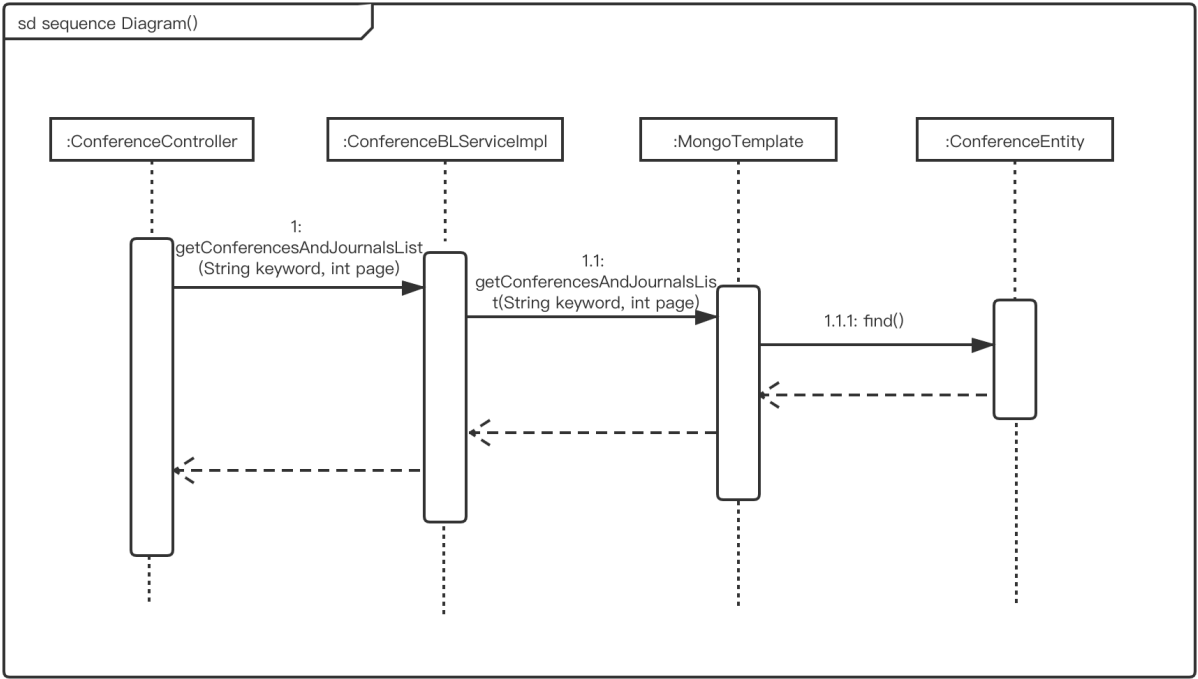


图14 获得会议/期刊列表的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

4.1.8 taskbl 模块

(1) 模块概述

task 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

task 模块的职责及接口参见软件系统结构描述文档表22。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.TaskService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

task 模块的设计如图7所示

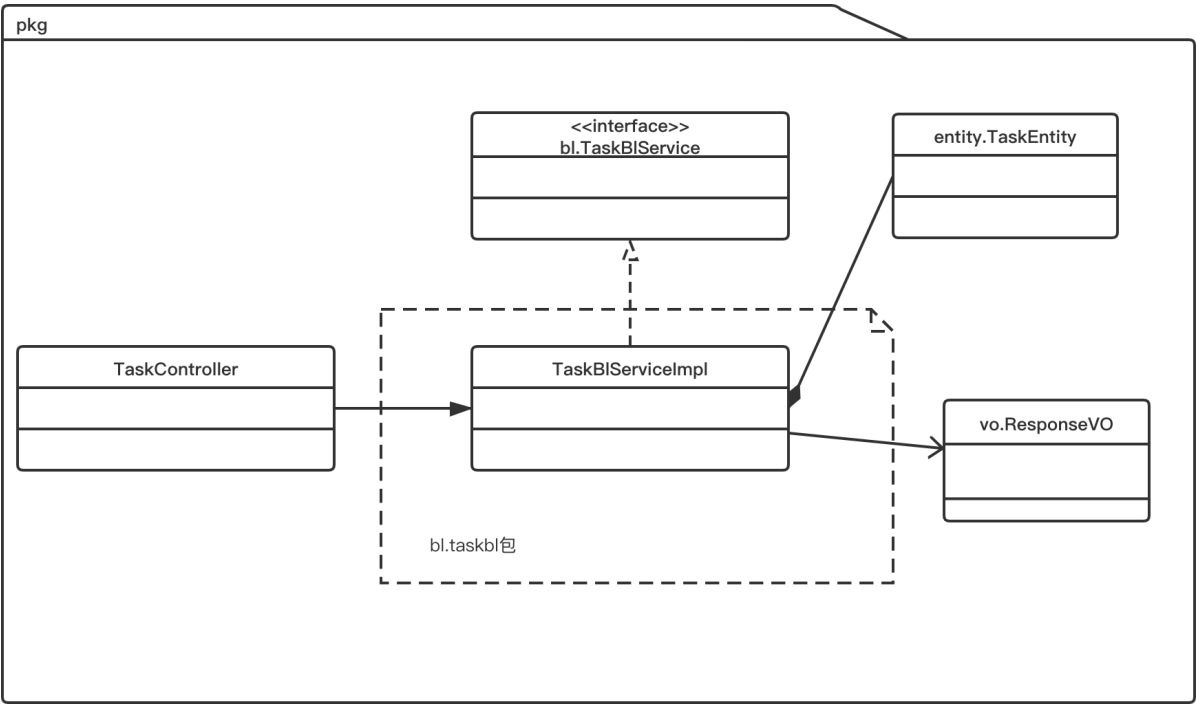


图15 taskbl模块各个类的设计

taskbl模块各个类的职责如表11所示：

表15 taskbl模块各个类的职责

模块	职责
TaskBLServiceImpl	负责实现爬取论文所需要的服务
TaskEntity	爬虫任务的领域模型对象

(3) 模块内部类的接口规范

TaskBLServiceImpl 的接口规范如表12所示：

表16 TaskBLServiceImpl模块的接口规范

表22 TaskBLService模块的接口规范		
提供的服务（供接口）		
TaskBLServiceImpl.getCrawlTask	语法	public BasicResponse getCrawlTask(String filterKey, Date date)
前置条件	传入符合条件的过滤标准和日期	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.find()	查找符合条件的结果集	

(4) 业务逻辑层的动态模型

图12表明了OASIS系统中，当用户发出获取某个日期的爬虫任务请求后，图谱相关业务逻辑处理的相关对象之间的协作。

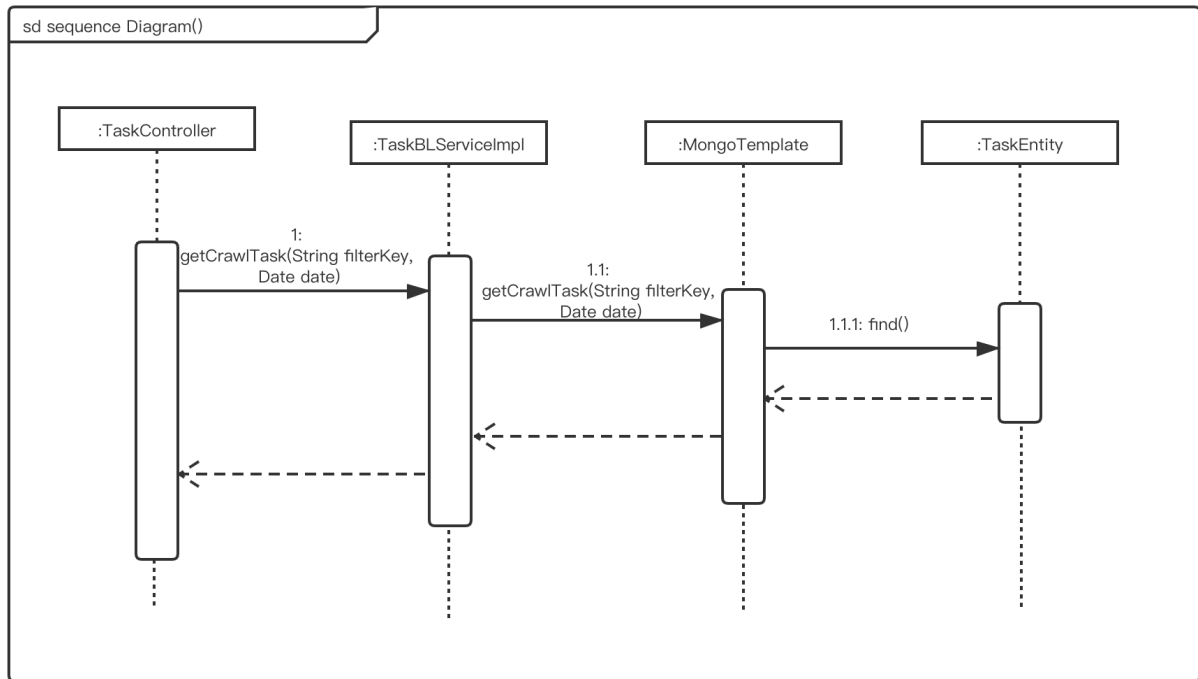


图16 获得某个日期爬虫任务列表的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

4.1.9 talentbl 模块

(1) 模块概述

talent 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

talent 模块的职责及接口参见软件系统结构描述文档表23。

(2) 整体结构

根据体系结构的设计，我们使用分层的模式来构建本系统，将系统分为web展示层、restful api层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如restful api层和业务逻辑层之间。我们添加bl.TalentService 接口。这样，我们将依赖于接口而非实体类，增加了系统的灵活性。

talent 模块的设计如图7所示

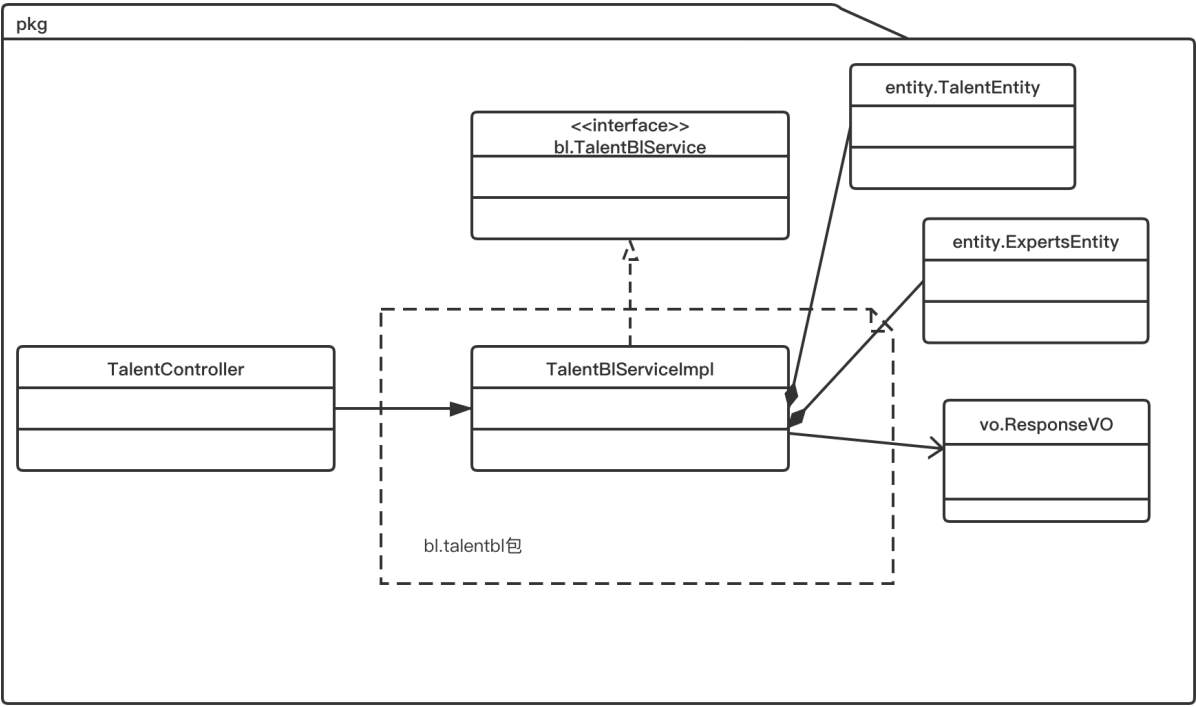


图17 talentbl模块各个类的设计

talentbl模块各个类的职责如表11所示：

表17 talentbl模块各个类的职责

模块	职责
TalentBServiceImpl	负责实现人才库所需要的服务
TalentsEntity	人才库的领域模型对象，拥有人才库的相关信息
ExpertsEntity	领域专家的领域模型对象，拥有领域专家的相关信息

(3) 模块内部类的接口规范

TalentBServiceImpl 的接口规范如表12所示：

表18 TalentBServiceImpl模块的接口规范

表23 TalentBLService模块的接口规范		
提供的服务（供接口）		
TalentBLServiceImpl.getActiveTalentsBase	语法	public BasicResponse getActiveTalentsBase();
前置条件	无	
后置条件	无	
TalentBLServiceImpl.getTalentsListByTalentBase	语法	public BasicResponse getTalentsListByTalentBase(String field, int page)
前置条件	传入符合条件的 领域和页数	
后置条件	无	
需要的接口（需接口）		
mongoTemplate.find()	查找符合条件的 结果集	
mongoTemplate.count()	计算符合条件的 结果集大小	

(4) 业务逻辑层的动态模型

图12表明了OASIS系统中，当用户发出查看某个领域的人才库请求后，相关业务逻辑处理的相关对象之间的协作。

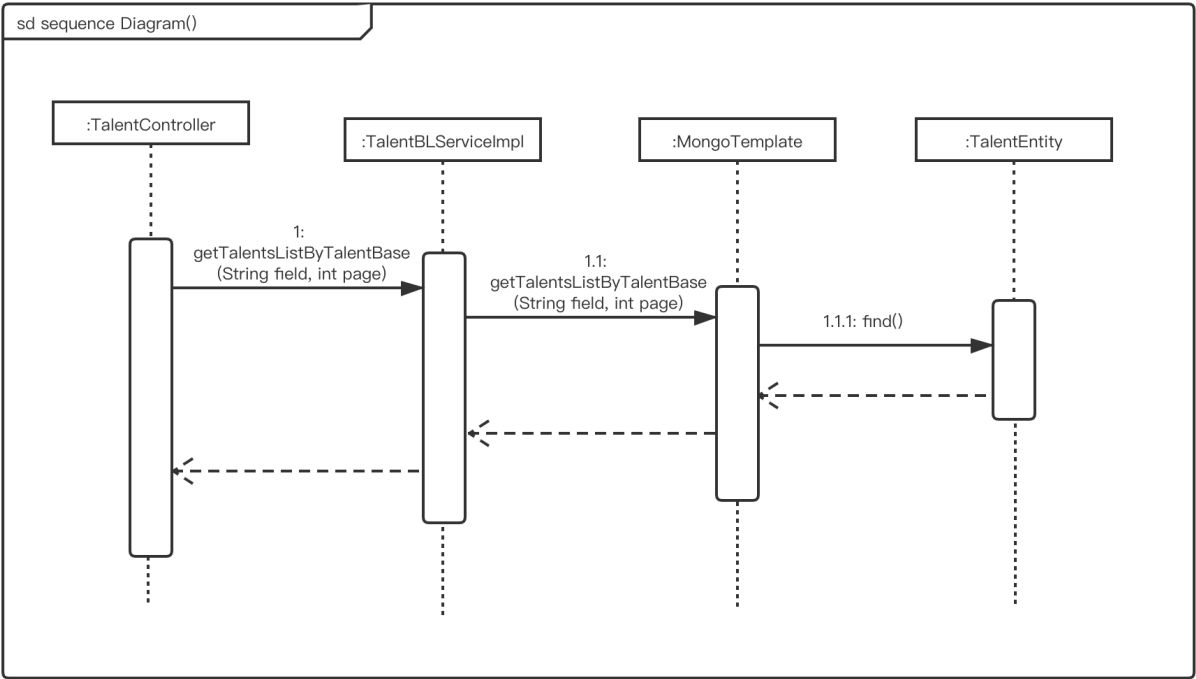


图18 获得某个领域的人才库请求的顺序图

(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的ServiceImpl委托给不同的领域对象。

5. 依赖视角

图9是服务器端包之间的依赖关系：

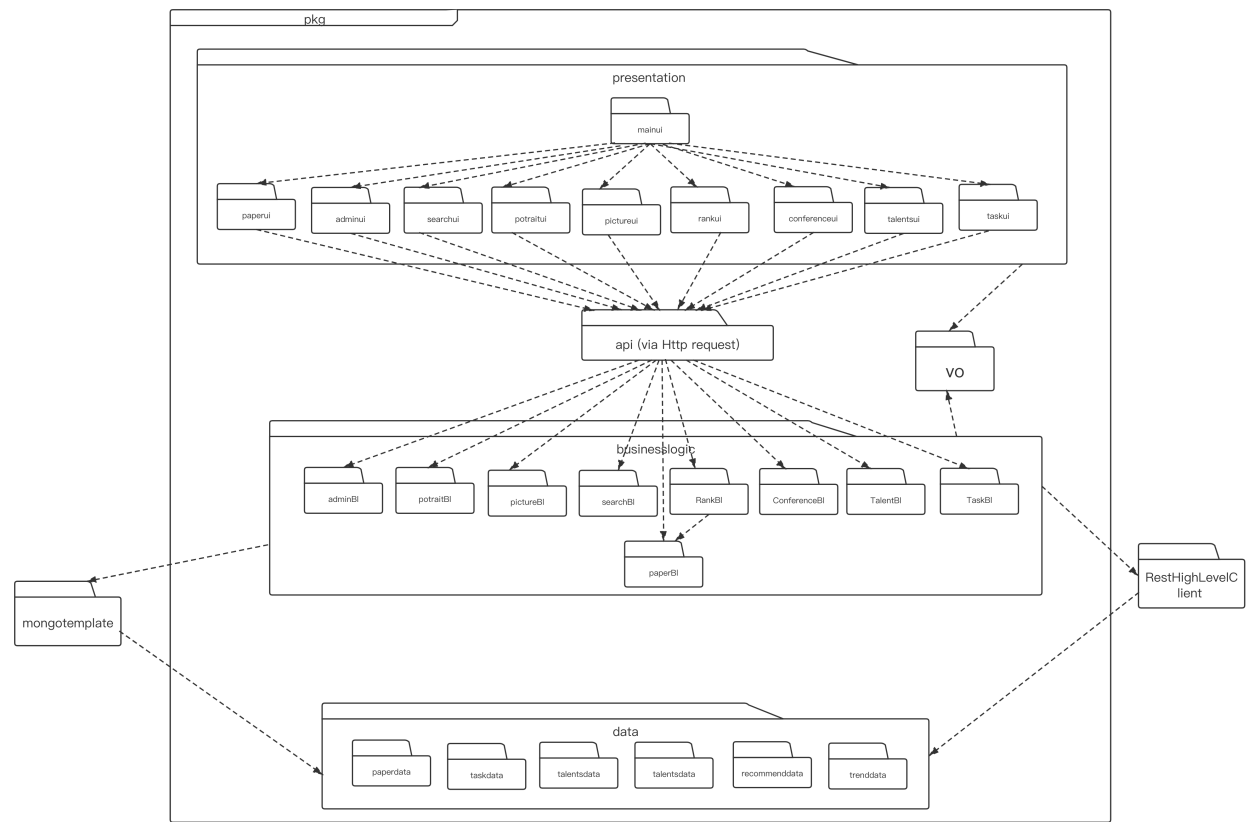


图19 服务器端包图