

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

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
 - 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12。

(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：

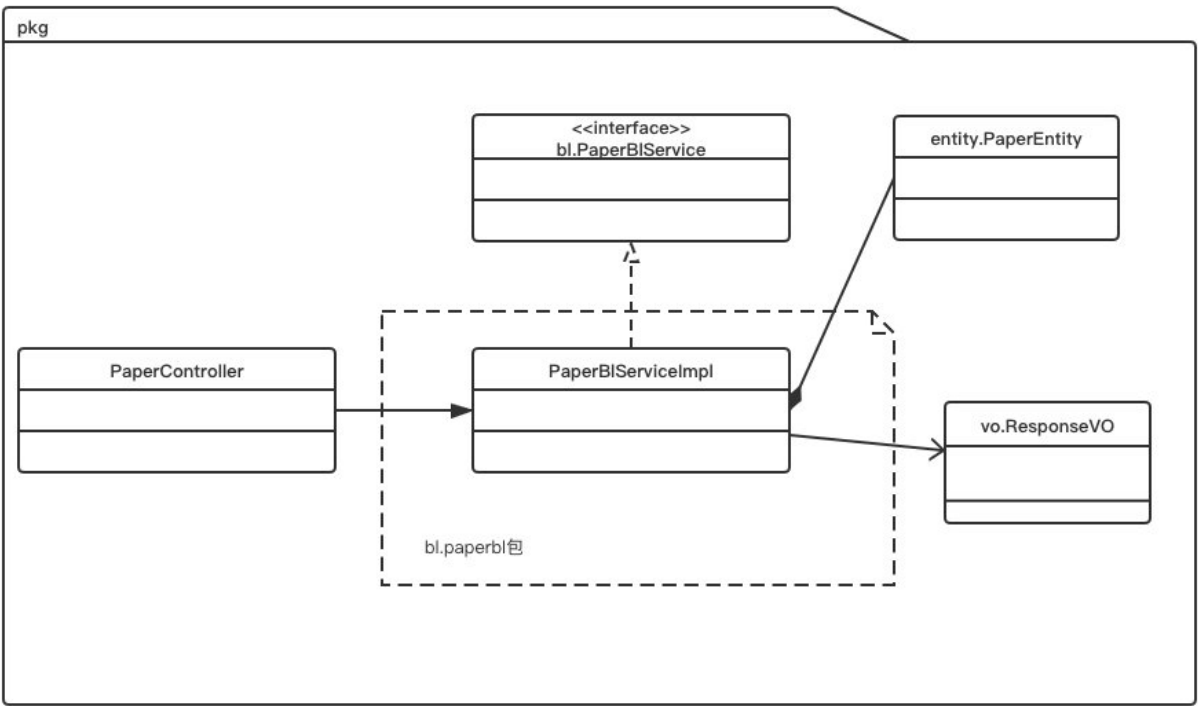


图1 paperbl模块各个类的设计

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
PaperBLServicImpl	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

PaperBLServicImpl 的接口规范如表2所示：

表2 PaperBLServicImpl的接口规范

提供的服务（供接口）		
PaperBLServicImpl.getResearcherInterest	语法	public BasicResponse getResearcherInterest(String id);
前置条件	无	
后置条件	无	

PaperBLServicImpl.getAffiliationInterest	语法	public BasicResponse getAffiliationInterest(String id)
前置条件	无	
后置条件	无	
PaperBLServicImpl.getJournalInterest	语法	public BasicResponse getJournalInterest(String id)
前置条件	无	
后置条件	无	
PaperBLServicImpl.getConferenceInterest	语法	public BasicResponse getConferenceInterest(String id)
前置条件	无	
后置条件	无	
PaperBLServicImpl.getActivePaperAbstract	语法	public BasicResponse getActivePaperAbstract();
前置条件	无	
后置条件	无	
PaperBLServicImpl.getReferenceById	语法	public BasicResponse getReferenceById(String paperId);
前置条件	无	
后置条件	无	
PaperBLServicImpl.getAuthorPapersById	语法	public BasicResponse getAuthorPapersById(String authorId, int page, String sortKey);
前置条件	无	
后置条件	无	
PaperBLServicImpl.getAffiliationPapers	语法	public BasicResponse getAffiliationPapers(String affiliation, int page, String sortKey);
前置条件	无	
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	
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。

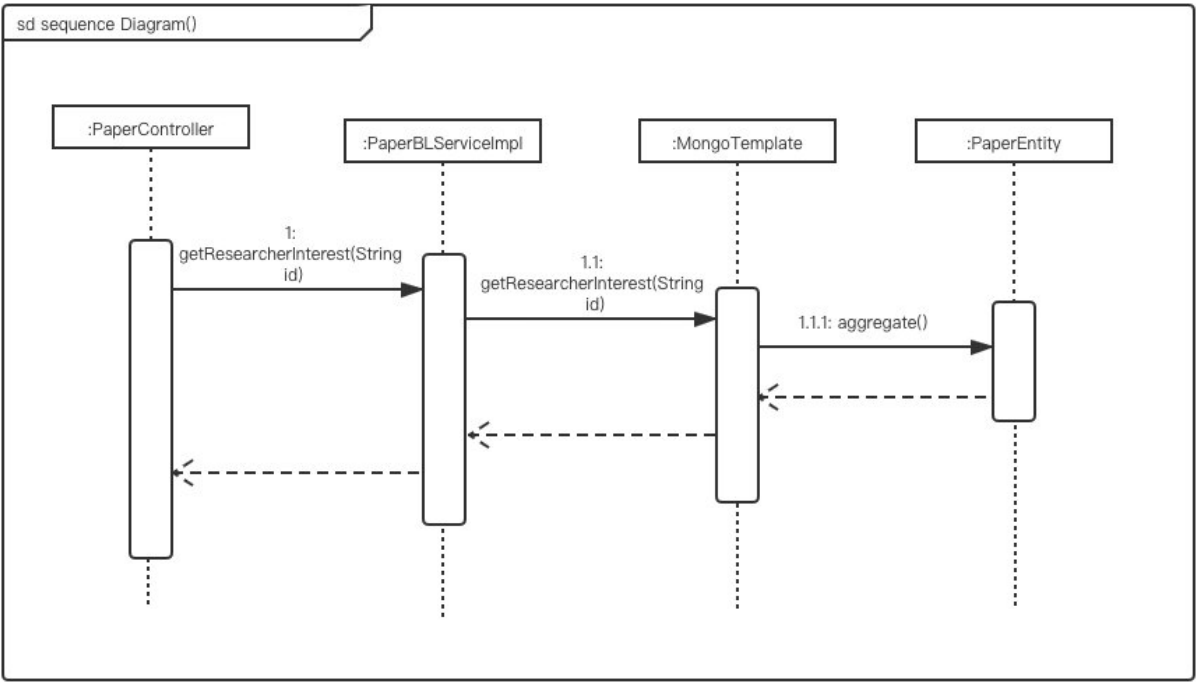


图2 得到学者研究方向的顺序图

(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BLServiceImpl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13。

(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：

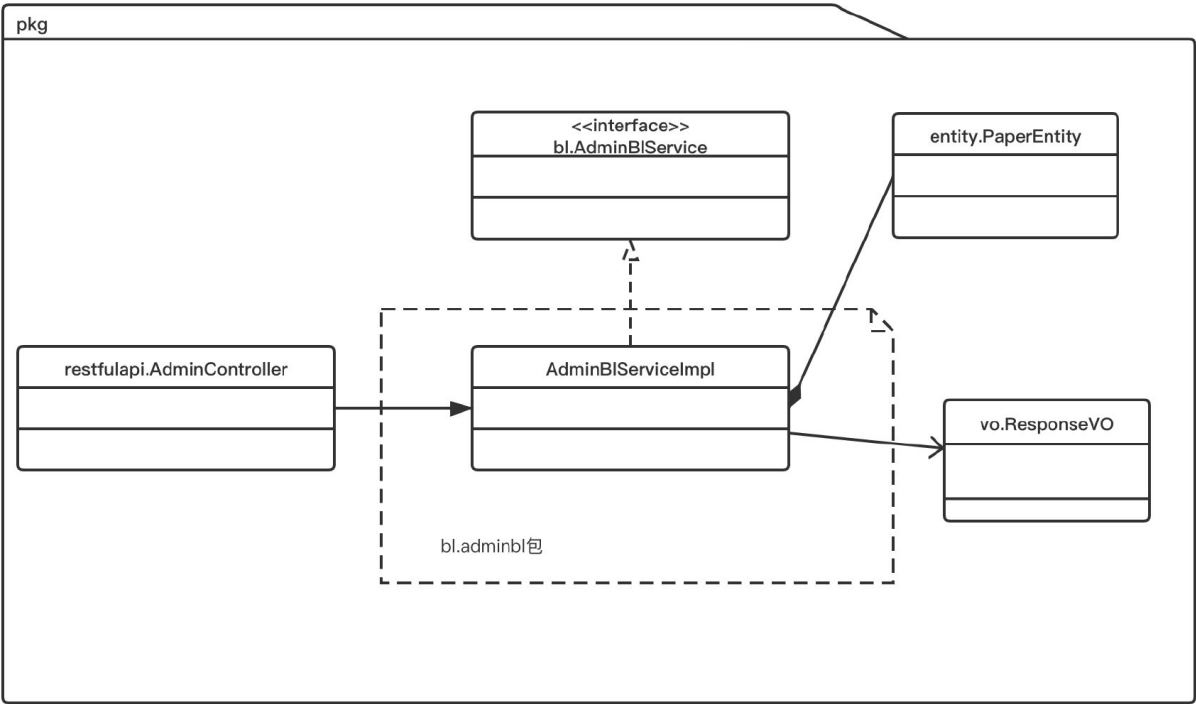


图3 adminbl模块各个类的设计

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
AdminBLServiceImpl	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

AdminBLServiceImpl 的接口规范如表4所示：

表4 AdminBLServiceImpl的接口规范

提供的服务（供接口）		
AdminBLSERVICEImpl.importPaperData	语法	public BasicResponse importPaperData(MultipartFile file);
前置条件	管理员已被授权	
后置条件	导入数据	
AdminBLSERVICEImpl.getConferenceInfo		public BasicResponse getConferenceInfo(int page, String name);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.getAffiliationInfo		public BasicResponse getAffiliationInfo(int page, String name);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.getJournalInfo		public BasicResponse getJournalInfo(int page, String name);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.getAuthorInfo		public BasicResponse getAuthorInfo(int page, String name);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.mergeAffiliationInfo		public BasicResponse mergeAffiliationInfo(List src, String desc);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.updateConferenceInfo		public BasicResponse updateConferenceInfo(String src, String desc);
前置条件	管理员已被授权	
后置条件	无	
AdminBLSERVICEImpl.updateJournalInfo		public BasicResponse updateJournalInfo(String src, String desc);
前置条件	管理员已被授权	
后置条件	无	
		public BasicResponse

AdminBLServicImpl.updatePaperInfo		updatePaperInfo(UpdatePaperParameter parameter);
前置条件	管理员已被授权	
后置条件	无	
AdminBLServicImpl.mergeAuthorInfo		public BasicResponse mergeAuthorInfo(List src, String desc);
前置条件	管理员已被授权	
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。

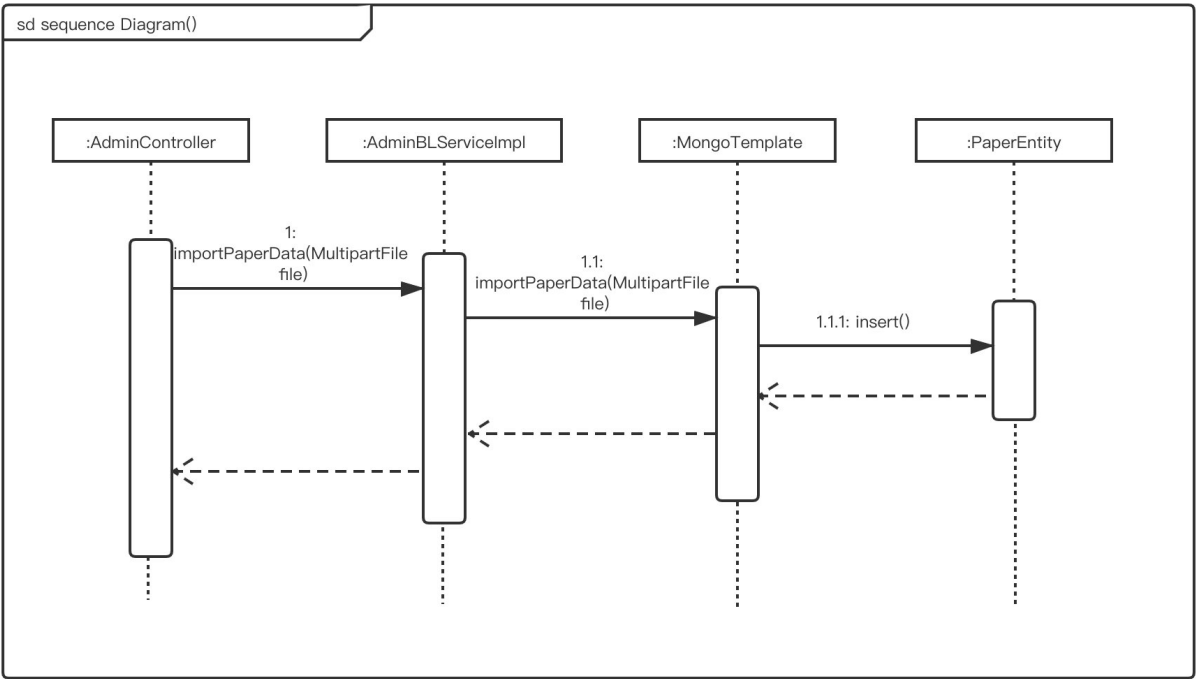


图4 导入论文的顺序图

(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BLServiceImpl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14。

(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.SearchBIService 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：

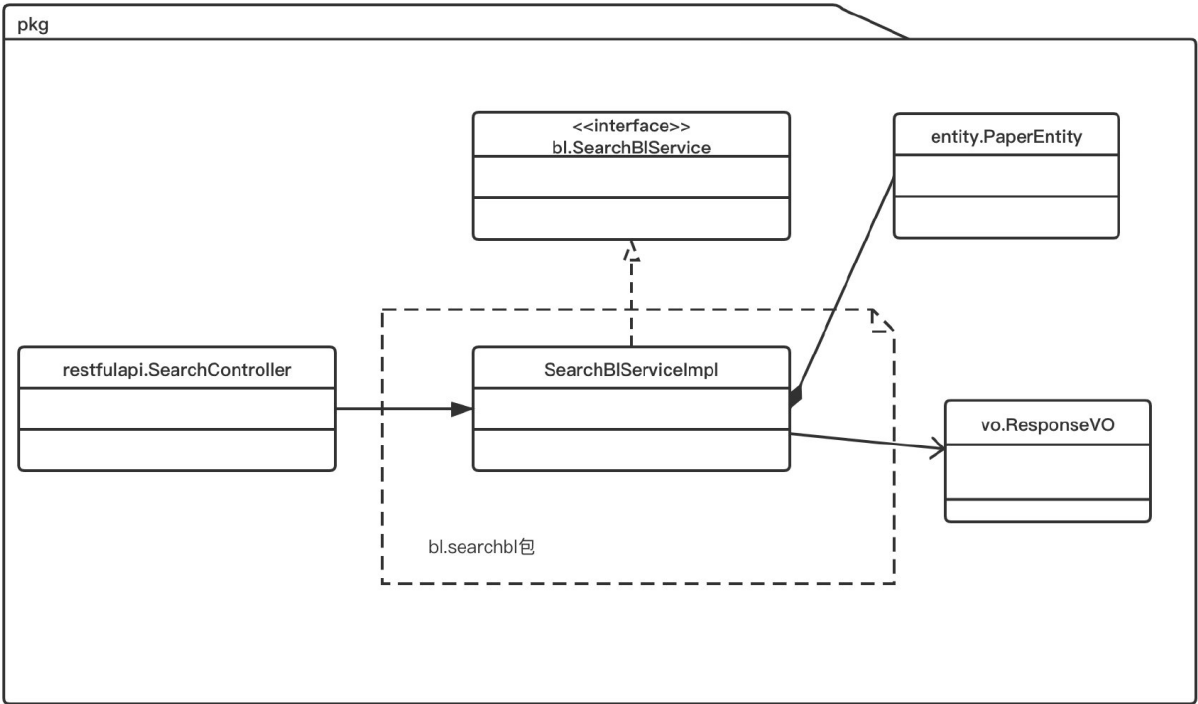


图5 searchbl模块各个类的设计

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	无	
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	无	
后置条件	无	
SearchBLSERVICEImpl.getBasicSearchFilterCondition	语法	public BasicResponse getBasicSearchFilterCondition(String keyword);
前置条件	无	
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	

(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。

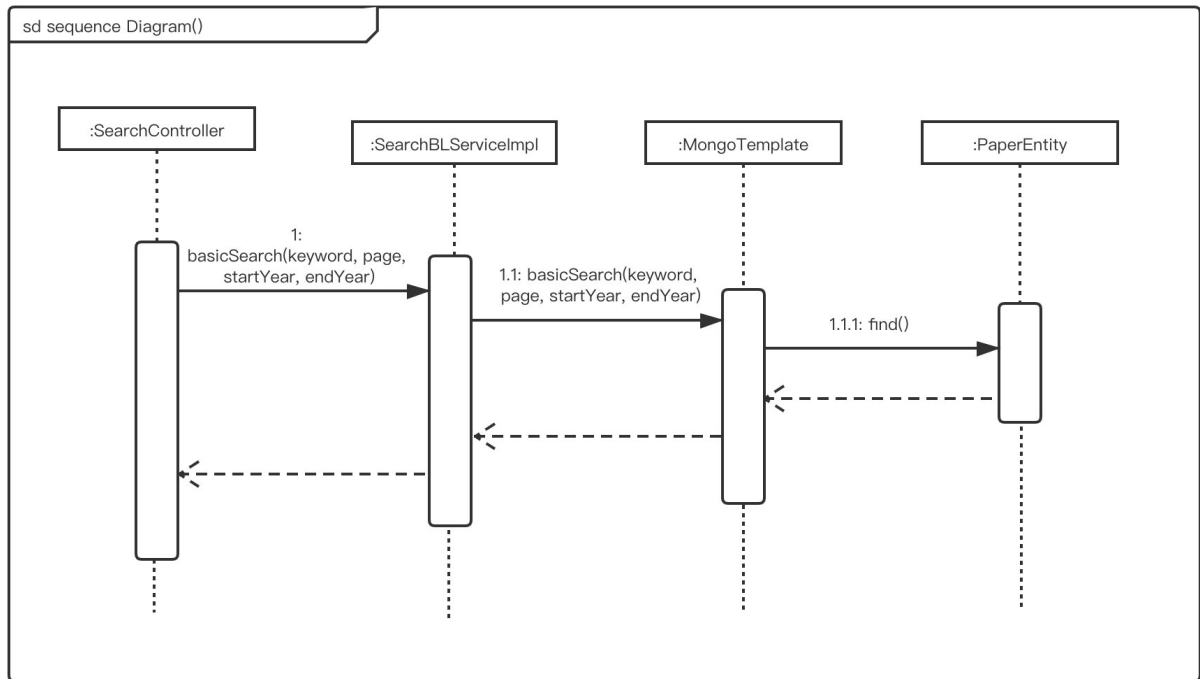


图6 基础查询的顺序图

(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BLSERVICEImpl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。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

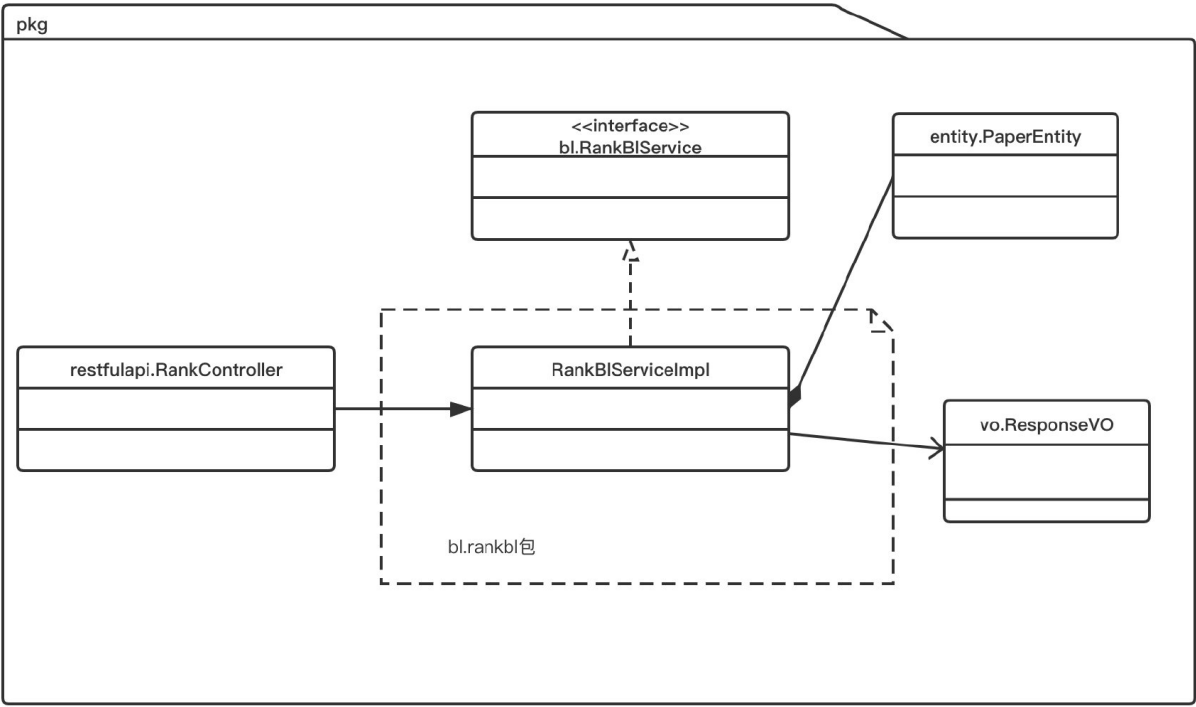


图7 rankbl模块各个类的设计

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	无	
后置条件	无	
RankBLSERVICE.getAuthorBasicRanking	语法	public BasicResponse getAuthorBasicRanking(String sortKey, String year);

前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getJournalBasicRanking		public BasicResponse getJournalBasicRanking(String sortKey, int year);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getConferenceBasicRanking		public BasicResponse getConferenceBasicRanking(String sortKey, int year);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getKeywordBasicRanking		public BasicResponse getKeywordBasicRanking(int year);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getAuthorAdvancedRanking		public BasicResponse getAuthorAdvancedRanking(String sortKey, int startYear, int endYear);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getAffiliationAdvancedRanking		public BasicResponse getAffiliationAdvancedRanking(String sortKey, int startYear, int endYear);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getAffiliationDetailRankingById		public BasicResponse getAffiliationDetailRankingById(String id);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getAuthorDetailRanking		public BasicResponse getAuthorDetailRanking(String affiliation);
前置条件	无	
后置条件	无	
RankBLSERVICEImpl.getAuthorDetailRankingById		public BasicResponse AuthorRankDetail getAuthorDetailRankingById(String id);

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。

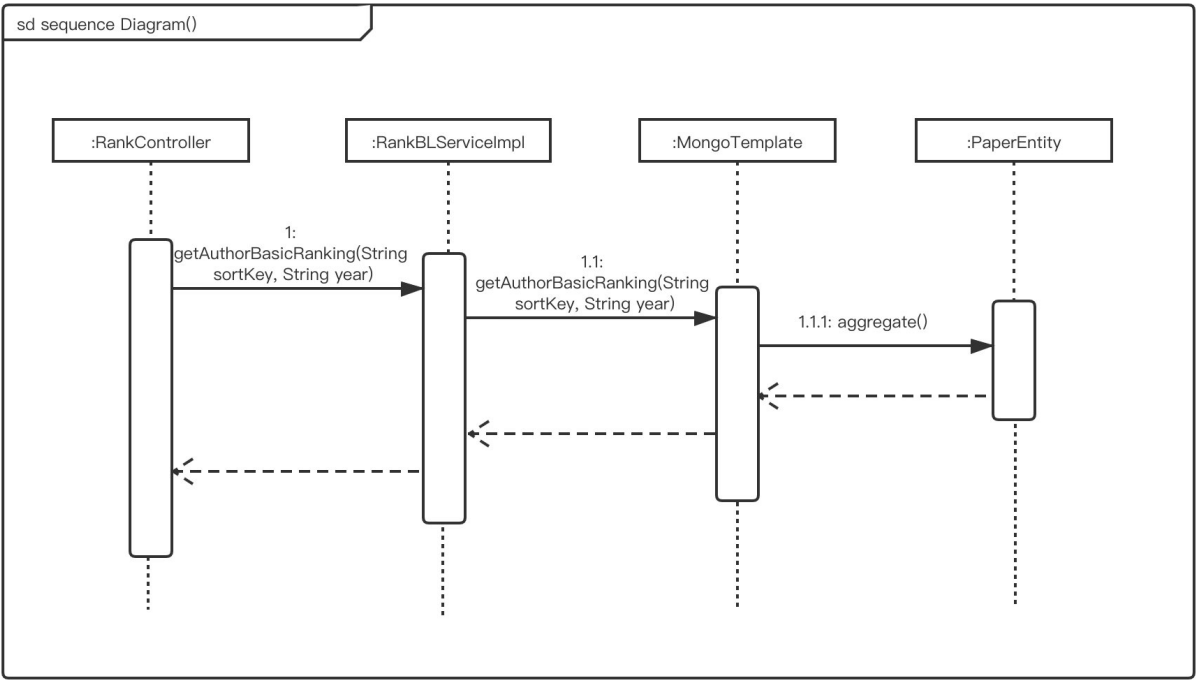


图8 获得作者排名的顺序图

(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13。

(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

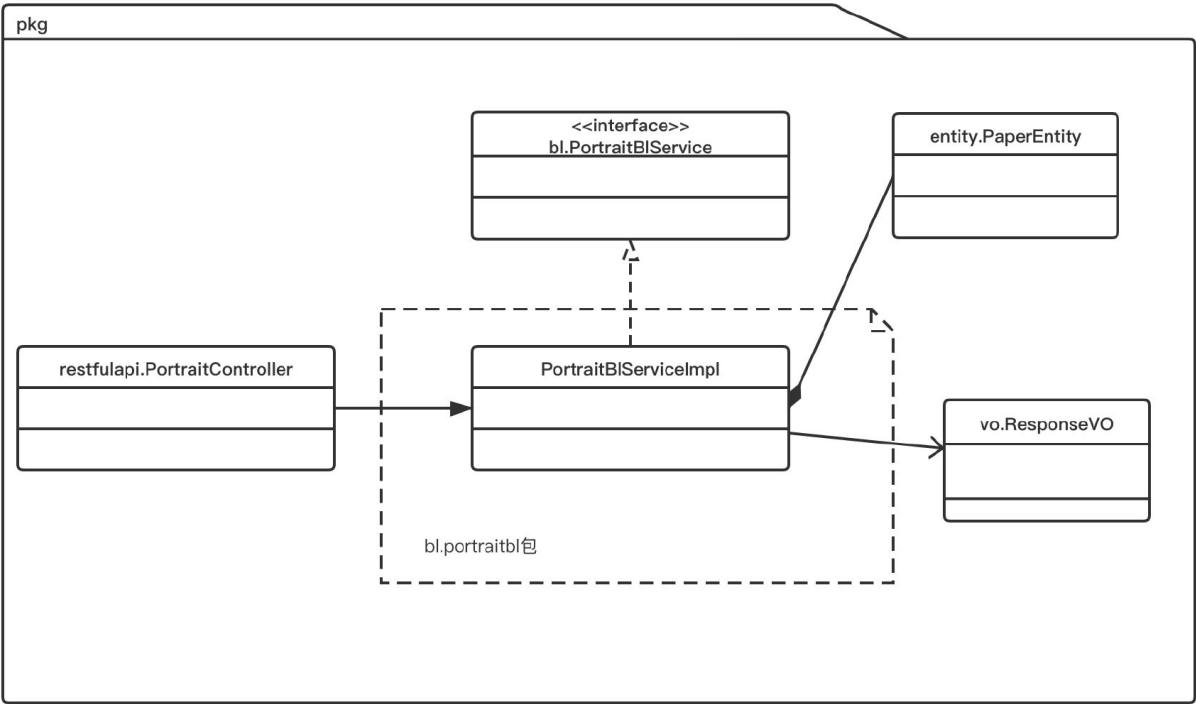


图9 portraitbl模块各个类的设计

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）		
PortraitBLServicImpl.getAuthorPortraitById	语法	public BasicResponse getAuthorPortraitById(String id);
前置条件	无	
后置条件	无	
PortraitBLServicImpl.getAffiliationPortrait	语法	public BasicResponse getAffiliationPortrait(String affiliation);
前置条件	无	
后置条件	无	
PortraitBLServicImpl.getAffiliationPortrait	语法	public BasicResponse getKeywordPortrait(String keyword);
前置条件	无	
后置条件	无	
PortraitBLServicImpl.getAffiliationPortrait	语法	public BasicResponse getAffiliationPortrait(String conference);
前置条件	无	
后置条件	无	
PortraitBLServicImpl.getAffiliationPortrait	语法	public BasicResponse getAffiliationPortrait(String journal);
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。

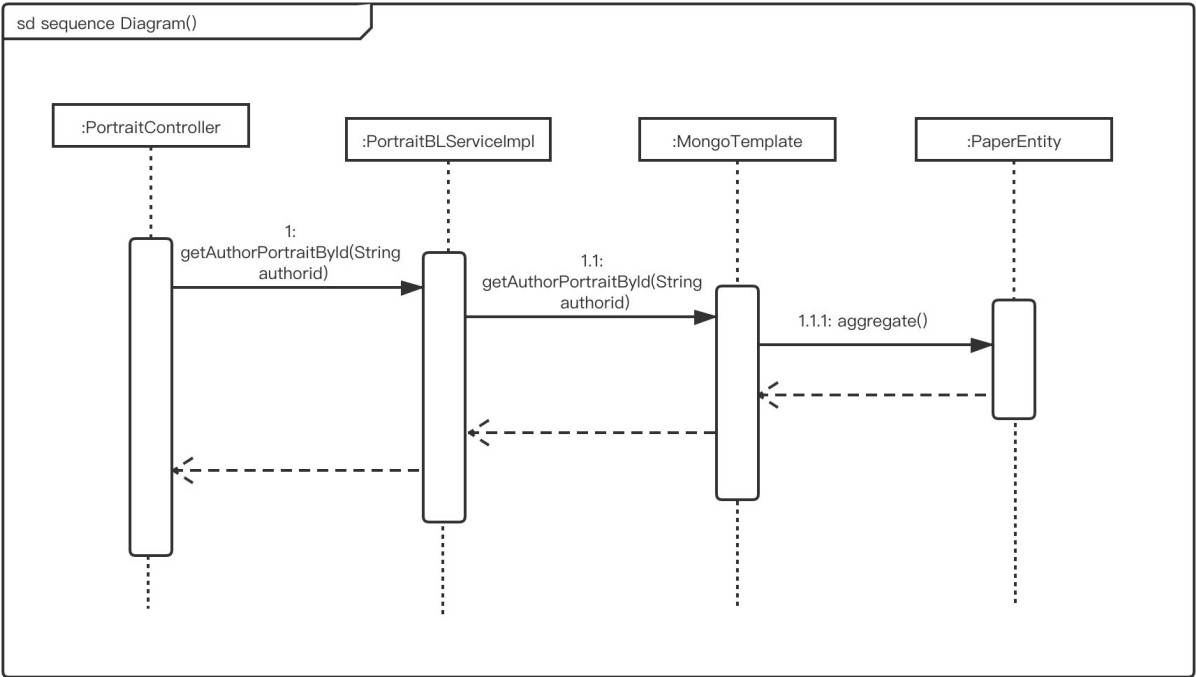


图10 获得作者用户画像的顺序图

(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.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

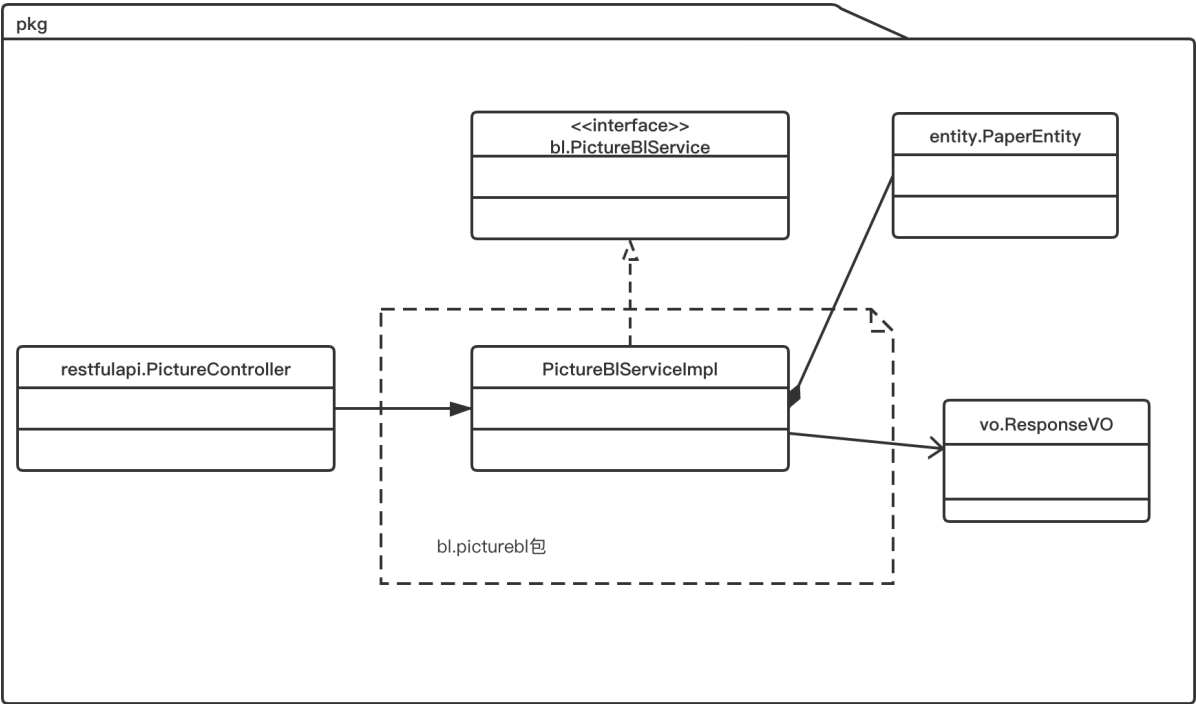


图11 picturebl模块各个类的设计

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	无	
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。

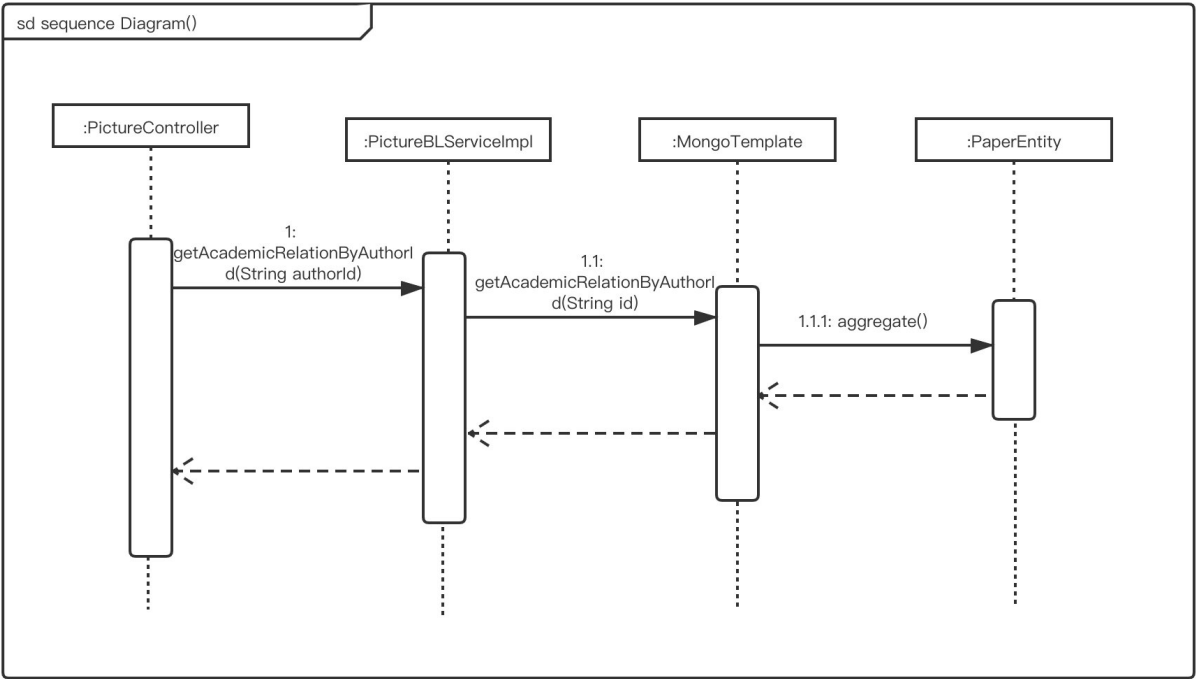


图12 获得作者排名的顺序图

(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。

5. 依赖视角

图9是服务器端包之间的依赖关系：

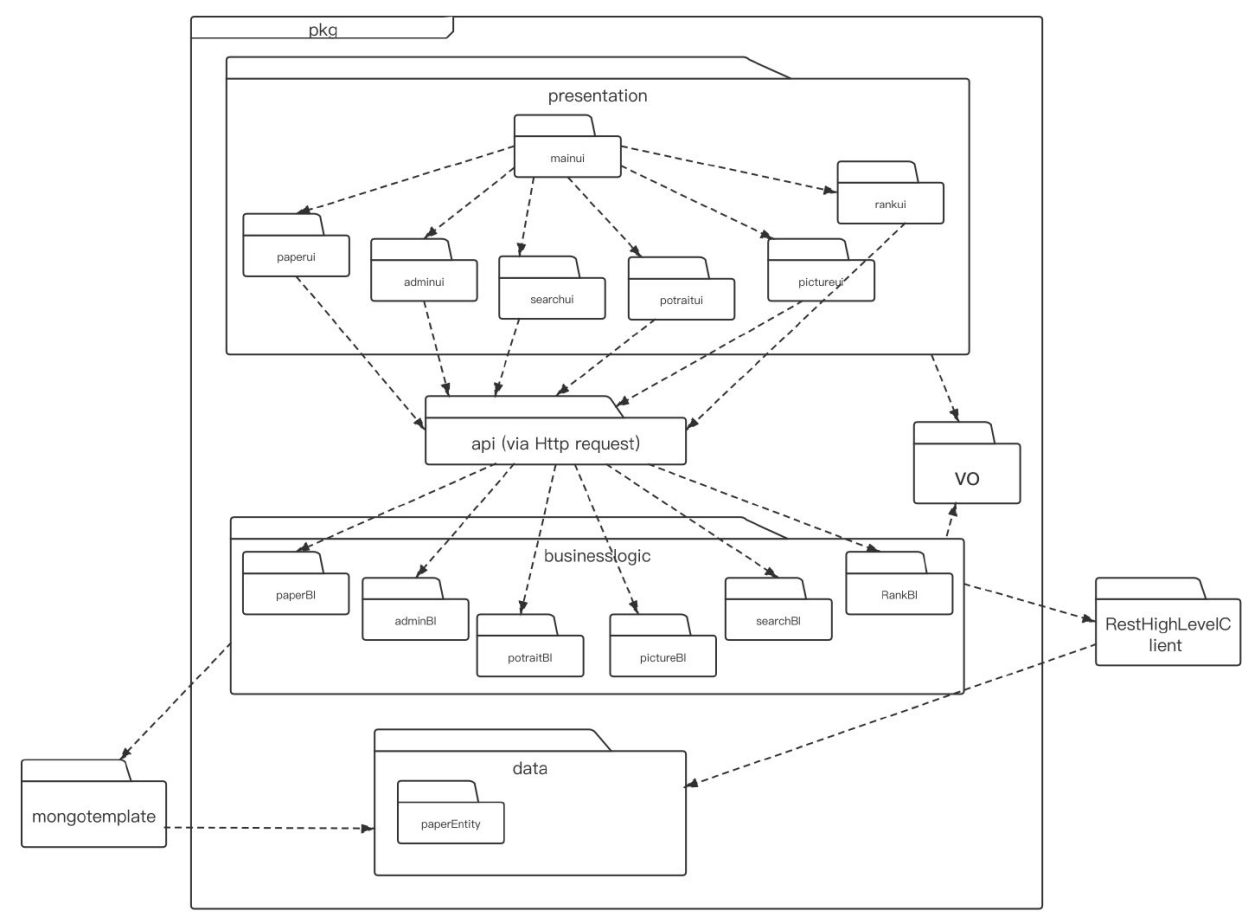


图13 服务器端包图