OMSE 555/556 Software Engineering Practicum I & II

Software Architecture Document

N-Canvas

C-Team-5

Version 1.0

10/20/2019

**更改历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **更改内容** | **更改人** | **日期** |
| 1.0 | 初始版本 | 屠昕远 | 10/20/19 |
| 1.1 | 填写用例视图 | 段皓铧 | 10/21/19 |
| 1.2 | 填写所有接口（功能） | 屠昕远 | 10/21/19 |

**Approval Block**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Comments** | **Responsible Party** | **Date** |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 1](#_Toc22629768)

[1.1. 目的 1](#_Toc22629769)

[1.2. 说明内容 1](#_Toc22629770)

[1.3. 写义与缩写 1](#_Toc22629771)

[1.4. 引用 1](#_Toc22629772)

[1.5. 概览 1](#_Toc22629773)

[2. 描述方法简介 2](#_Toc22629774)

[3. 用例视图 2](#_Toc22629775)

[3.1. 作用 2](#_Toc22629776)

[3.2. 部分用例示意 2](#_Toc22629777)

[3.2.1. 登录 2](#_Toc22629778)

[3.2.2. 课程 3](#_Toc22629779)

[3.2.3. 作业 4](#_Toc22629780)

[3.2.4. 邮件 4](#_Toc22629781)

[3.2.5. 讨论 5](#_Toc22629782)

[4. 逻辑视图与数据流向 6](#_Toc22629783)

[4.1. 概述 6](#_Toc22629784)

[4.2. 接口定义 6](#_Toc22629785)

# 引言

## 目的

本文档在较高的层次上提供了概述，并说明了N-Canvas系统的体系结构。

该文档说明了体系结构的目标，系统支持的用例和体系结构样式等系统架构方面的设计。该文档为决策设计和体系结构的构思与实现提供了理论依据。

软件体系结构文档（SAD）提供了N-Canvas系统的全面体系结构概述。为了尽可能准确地描述软件，本文档将分别采用用例视图与设计视图进行解释与说明。

## 说明内容

本SAD的解释的内容是N-Canvas系统的体系结构。

本文档描述了在N-Canvas系统设计中对体系结构较为重要的各个方面。这些要素和行为对于指导系统的构建以及从整体上理解本项目至关重要。

## 写义与缩写

* **Apache** – 网络服务器
* **HTTP** – 超文本传输协议
* **WWW** – 互联网
* **SAD -** 软件架构文档
* **UML** – 统一建模语言
* **User** - 使用本系统的已注册用户

## 引用

[SRS]: 软件需求规格说明书

[MedBiquitous]: SAD模板, http://medbiq.org/std\_specs/techguidelines/softwarearchitecture.pdf

## 概览

为了完整记录体系结构的所有方面，软件体系结构文档包含以下小节。

第2节：介绍每种视图

第3节：描述主要的用例

第4节：描述系统的逻辑视图，包括接口和操作定义

# 描述方法简介

本小节主要介绍了后面所用到的四种视图面向的读者与该视图所描述的内容。

**用例视图**

**面向读者**: 系统的所有利益相关者，包括最终用户。

**描述领域**: 描述了代表系统某些重要的中央功能的一组场景或用例。描述了系统的参与者和用例，此视图提出了用户的需求，并在设计级别上进行了详细说明，用以更详细地描述离散的流程和约束。该部分的词汇表与任何处理模型或表示语法无关。

**设计视图**

**面向读者**: 设计者。

**描述领域**: 设计视图是由用例图发展而来，是将需求转化为代码的前提。设计视图表示了设计模型中在构架方面具有重要意义的部分，即类、子系统、包和用例实现的子集。

# 用例视图

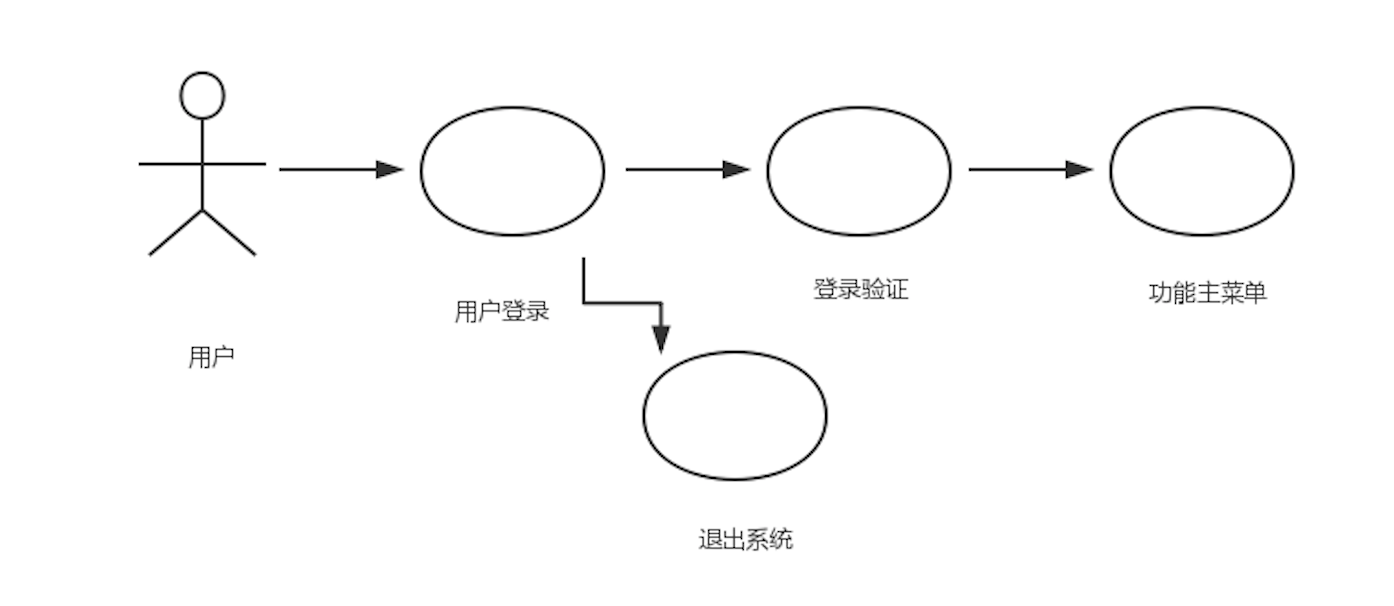
## 作用

用例视图的目的是为系统的使用及其组件之间的交互提供额外的上下文。 在本文档中，每个组件都被视为用例参与者。在第3.2节中，使用UML用例图和序列图概述和说明了最常见的用例，以阐明组件之间的交互。

## 部分用例示意

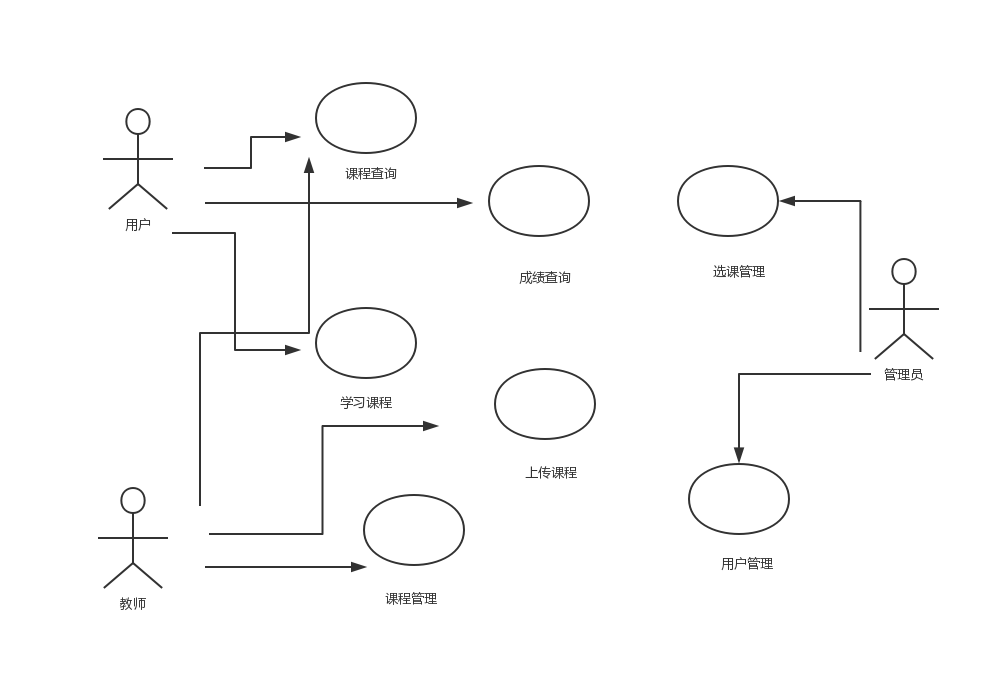
### 登录

**Figure 4.1** 登录用例图



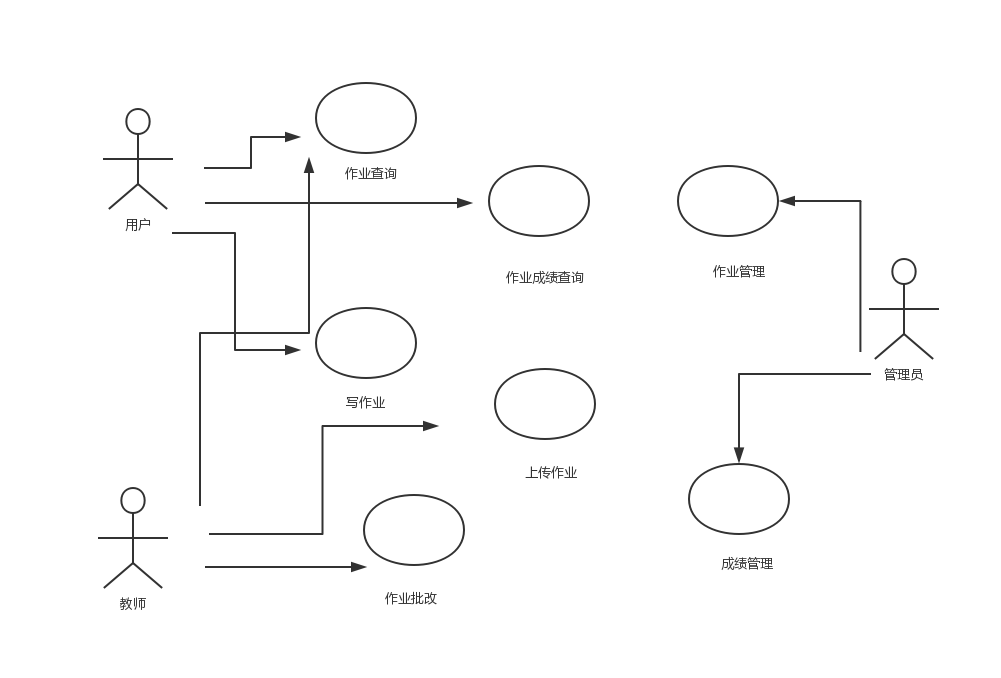
### 课程

**Figure 4.2** 课程用例图



### 作业

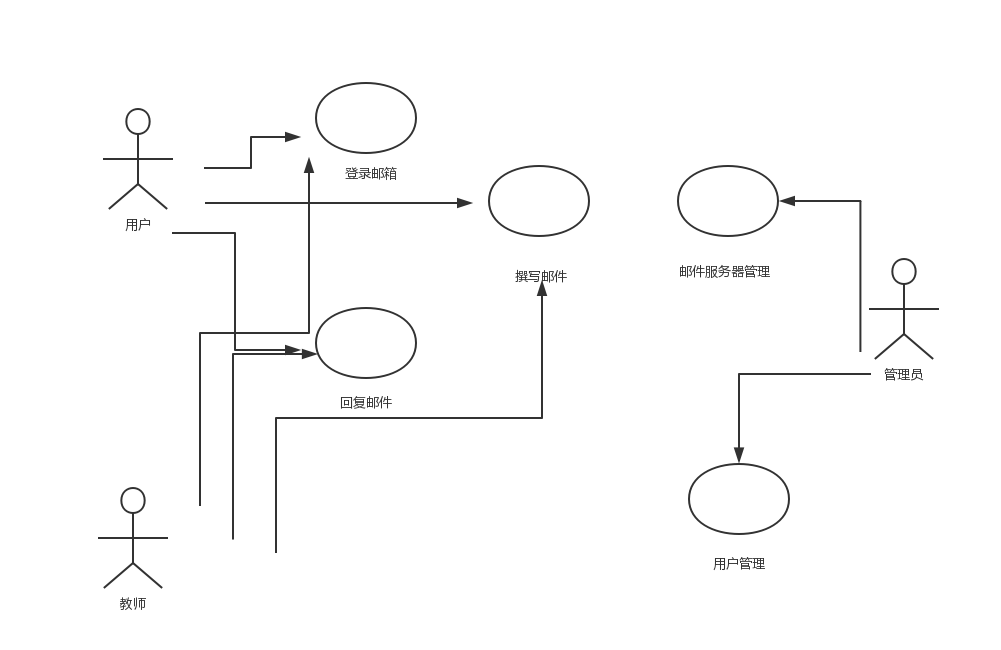
**Figure 4.3** 作业用例图



### 邮件

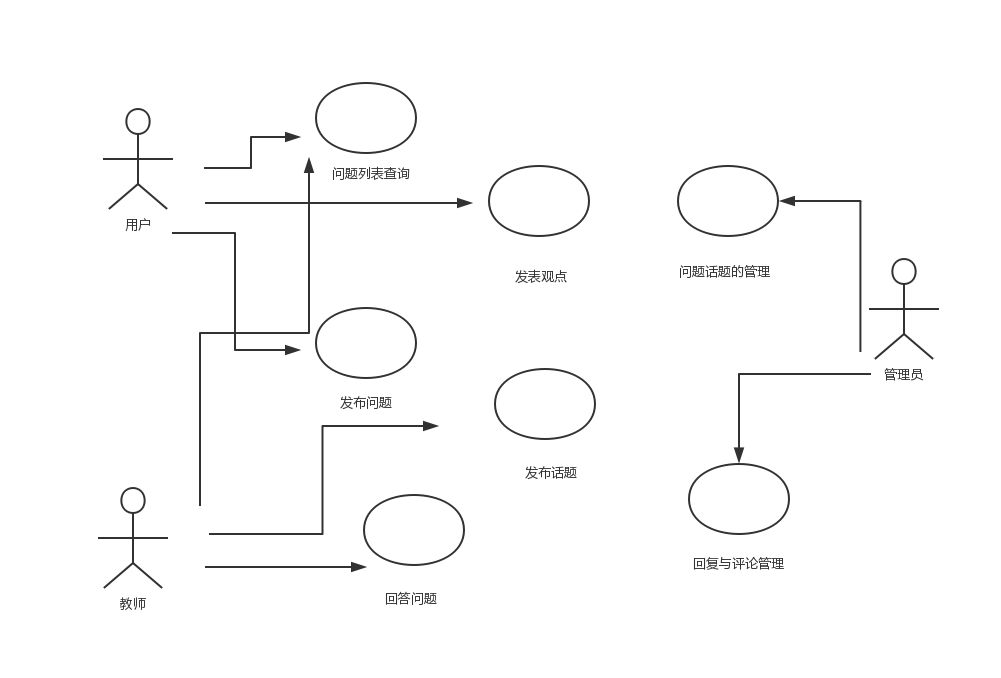
.

**Figure 4.4** 邮件用例图



### 讨论

**Figure 4.5** 讨论用例图



# 逻辑视图与数据流向

## 概述

逻辑视图的主要目标是定义组成系统的组件，并定义它们将通过它们相互通信和交互的接口。定义系统组件背后的主要决策因素是需要将可能发生变化的组件与系统其余部分隔离开来。通过清楚地定义这些组件的接口并将它们的内部实现对系统的其余部分隐藏，可以将预期更改的影响最小化。

本系统的数据流向主要有两条：用户端向服务器提交查询请求，服务器查询数据库后返回数据给用户端并显示；用户端向服务器提交数据（或改动请求），服务器将数据（或改动）应用于数据库中。因此，本节只需定义服务器中用以访问、更改数据库的类即可。

## 接口定义

## 登录：传入用户名与密码，返回是否存在该用户及用户类别

## 添加用户：传入用户名、密码与用户类型等，新建一个用户

## 删除用户：管理员传入用户名，删除一个用户

## 添加课程：传入课程的名字、授课老师等信息，添加一个课程

## 更改课程：传入要更改的课程的名字，以及要更改的信息（如课程时间等），更新数据

## 学生选课：管理员输入学生与课程名，帮助学生选课

## 设置或取消星标：输入学生与课程名，设置或取消星标课程

## 获得课程概览：输入课程名称，得到课程概览

## 上传课程概览：输入课程名称与课程概览文本，进行更新

## 获得课程大纲：输入课程名称，得到课程大纲

## 上传课程大纲：输入课程名称与课程大纲文本，进行更新

## 获得模块信息：输入课程名称，得到模块信息

## 更改模块信息：输入课程名称与模块信息，进行更新

## 获得单元内容：输入课程名称与单元号，得到单元内容

## 上传单元内容：输入课程名称、单元号和单元内容文本，进行更新

## 上传文件：输入一个文件名称与文件，记录

## 下载文件：输入一个文件名称，得到文件

## 获得作业：输入课程与作业名，得到作业内容（老师可以输入学生名获得学生的答案）

## 发布作业：输入课程与作业名和作业内容，进行记录

## 提交作业：输入课程名、作业名、学生名和做的作业内容，进行提交

## 批改作业：输入课程名、作业名、学生名和分数，进行记录

## 发布话题：输入课程名、模块号及话题标题内容，创建一个话题

## 删除话题：输入话题号并删除

## 评论：输入话题号、评论人、被评论的评论号及评论的信息，进行评论

## 删除评论：输入评论号并删除

## 获得邮件列表：输入用户名，得到所有收发邮件

## 获得邮件内容：输入邮件号，得到邮件内容

## 发邮件：输入用户名、发送目标用户名和邮件标题内容等，进行发送

## 获得帮助文档：返回所有（或选定的）帮助文档