

# 天津大学



## 天津大学融合门户系统分析与设计

学院： 智能与计算学部

班级： 2022 计科 3 班

姓名： 陈秋澄

学号： 3022244290

2024 年 10 月 19 日

## 目录

1 引言 .....	3
2 需求分析 .....	3
2.1 功能需求描述 .....	3
2.1.1 用户需求 .....	3
2.1.2 系统需求 .....	4
2.1.3 用例图 .....	5
2.1.4 非功能需求 .....	8
2.2 软件需求优先级及开发安排 .....	9
2.3 用例交互模型 .....	10
2.4 分析类模型 .....	12
2.5 状态模型 .....	14
3 体系结构设计 .....	16
3.1 体系结构的逻辑视图 .....	16
3.1.1 包图 .....	16
3.1.2 构件图 .....	19
3.2 体系结构的部署视图 .....	21
3.2.1 部署架构描述 .....	22
3.2.2 部署图 .....	22
3.2.3 关键技术及安全策略 .....	23
4 详细设计 .....	24
4.1 用例设计 .....	24
4.1.1 用例 1：用户登录 .....	24
4.1.2 用例 2：用户与办事大厅的交互 .....	25
4.1.3 用例 3：社团活动审批业务办理 .....	27
4.1.4 设计类图 .....	28
4.2 类设计 .....	29
4.2.1 用户类设计 .....	30
4.2.2 服务类设计 .....	33
4.3 数据设计 .....	34
4.3.1 数据库表设计 .....	34
4.3.2 数据库表结构示例 .....	34

4.4 子系统与构件设计 .....	37
4.4.1 系统整体架构 .....	37
4.4.2 子系统设计 .....	37
4.4.3 交互流程 .....	39
4.4.4 数据库设计 .....	39
5 软件测试 .....	39
5.1 单元测试 .....	40
5.1.1 社团活动审批服务 .....	40
5.1.2 绿化类维修申请服务 .....	42
5.2 确认测试 .....	44
5.2.1 登录功能的测试 .....	44
5.2.2 资讯中心功能的测试 .....	45
5.3 非功能测试 .....	47
5.3.1 并发性能测试 .....	47
5.3.2 配置和兼容性测试 .....	48
5.3.3 安全性测试 .....	49
5.4 测试总结 .....	50
6 项目总结 .....	51
6.1 系统设计的成效与不足 .....	51
6.1.1 系统设计的成效 .....	51
6.1.2 系统设计的不足 .....	51
6.2 未来改进方向 .....	52
6.3 个人总结 .....	53

# 1 引言

随着信息化建设的快速推进，高校对于信息管理平台的需求日益增长。天津大学融合门户系统作为学校的重要信息门户，集成了多个不同功能的子系统，为全校师生及工作人员提供了统一的信息入口和操作界面。该系统不仅覆盖了事务管办理、资讯与公告查询等日常功能、整合了图书馆、个人网盘、教务系统、大创系统、本科毕设系统、办事大厅、协同办公等关键应用服务，最重要的是实现了包括本科学生证补办、绿化类设施报修、社团活动审批等 41 项常用业务线上办理，实现了校内信息的集中化管理和跨部门数据的无缝对接。

本次设计报告通过分析天津大学融合门户系统，结合软件工程课程的理论和方法，完成对该系统的需求分析、体系结构设计、详细设计及测试方案的撰写，并对整个系统开发设计及现状进行总结与反思。

## 2 需求分析

### 2.1 功能需求描述

#### 2.1.1 用户需求

##### 1. 普通用户（学生及教职工）

###### (1) 首页快捷办公

登录融合门户首页，可以定制“我的应用”部分：即在“全部应用中”选取个人的常用网站（包括各学院官网、国际处、教育教学信息管理系统等），并将其拖动至“我的应用”部分即可直接跳转至常用的网站，避免了前去每个栏目逐个查找的操作，大大提高办事效率、提升用户的满意度。

###### (2) 资讯阅览及查找

资讯中心模块实现公告分类查询，菜单栏包括：校内动态、参考信息、单位动态、部门信息、本科教学、社科信息等。实现搜索功能：可根据标题、消息来源、发布时间对公告进行快速查找。

###### (3) 办理业务

办事大厅可办理材料传递、学费缓交申请、社团活动审批、本科生补办学生证等业务，并可查询本用户办理的事项；同时可在服务列表中

选择所属机构、服务分类、服务角色、项目名称，以快速筛选心仪服务。

(4) 应用查询

查询全部的校内网站，点击某个网站后并直接跳转。

(5) 协同办公

日程模块中可在个人日历中选取时间段标记个人事务，可将其导出共享。

## 2. 管理员

(1) 用户管理：管理员需要能够管理系统用户，包括添加、删除用户，分配角色和权限。

(2) 公告管理：管理员希望能够创建、编辑、发布或删除校园公告，确保重要信息能够及时传达给不同用户群体。

(3) 系统维护：管理员需要执行系统的维护操作，包括数据备份和恢复，确保系统在出现问题时可以快速修复。

(4) 权限分配：管理员希望能够灵活配置系统的访问权限，确保不同角色的用户只能访问对应的功能模块。

### 2.1.2 系统需求

#### 1. 用户身份验证

用户身份验证是门户系统的基础功能，确保系统的安全性以及用户角色的区分。系统要求所有用户通过统一身份认证系统（Single Sign-On, SSO）进行身份验证。通过输入学工号与密码，系统根据登录凭据识别用户身份，并为学生、教师和管理员分配相应的权限。

用户输入正确的用户名和密码后，系统应立即验证用户身份并跳转至用户的融合门户首页。若登录失败，系统应提示用户失败原因，如“密码错误”或“账户不存在”。

#### 2. 业务办理与公告管理

首先，办事大厅的业务办理服务的权限可根据用户身份进行分类展示。如：学生申请借用教室、补办学生证；教师则可批阅与其所管理社团或业务相关的申请或审批、查看与教学科研相关的公告；工作人员可以查看所属部门的报修信息并更新进度或状态等等。

其次，门户系统需要提供一个公告与通知的展示平台，帮助校方发布校园重要信息，通知学生和教师校历更新及校园活动。

该功能还要求管理员能够创建、编辑、发布或删除公告，保证信息的实时性与有效性。

此外，公告与办事大厅需具有时间戳功能，以能够自动显示公告发布时间、及业务办理环节，以为公告的按时间搜索功能和事项办理流程的查看提供基础。

### 3. 系统管理功能

系统管理员负责系统的维护，包括用户管理、数据备份与恢复等。管理员可以对学生和教师用户进行权限分配、角色修改，确保不同类型的用户能够正确访问其权限内的功能。此外，管理员可以发布系统公告，定期备份系统数据，确保系统稳定运行。

## 2.1.3 用例图

为了清晰展示用户与系统之间的功能交互关系，依据总体、办事大厅、某一服务三个层级逐一使用用例图对不同用户群体的操作进行建模。

### 1. 第一层级——总体层面

#### (1) 用例图

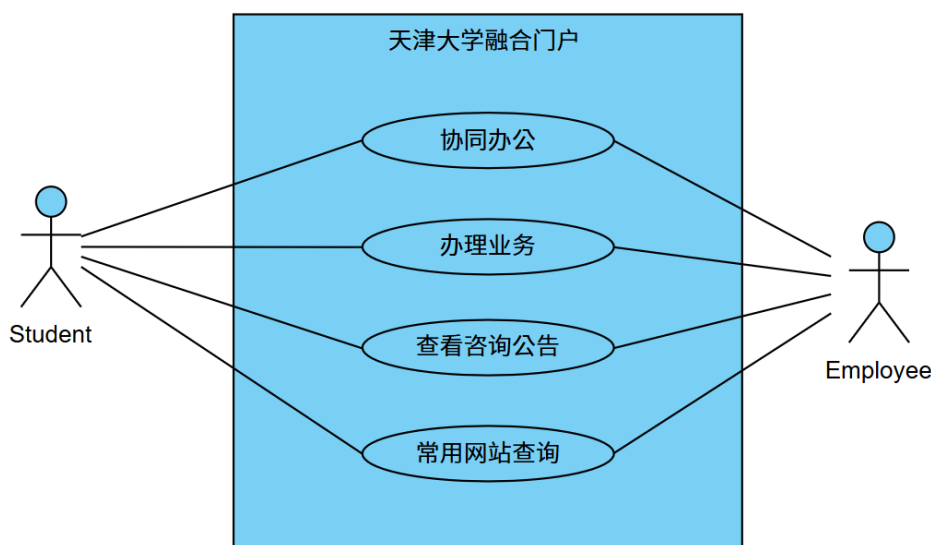


图 2-1 天津大学融合门户用例图（总体层级）

#### (2) 详细描述

- a) 用例名：融合门户系统登录；
- b) 用例标识：UC - ConvergencePortal；
- c) 主要执行者：学生，教职工；
- d) 目标：通过合法身份登录门户系统以获得不同的操作权限；
- e) 范围：天津大学融合门户系统；

- f) 前置条件：使用融合门户系统之时；
- g) 交互动作：用户输入学工号和密码；系统验证用户账号和密码的正确性和合法性；验证正确则意味着登录成功。

## 2. 第二层级——办事大厅

### (1) 用例图

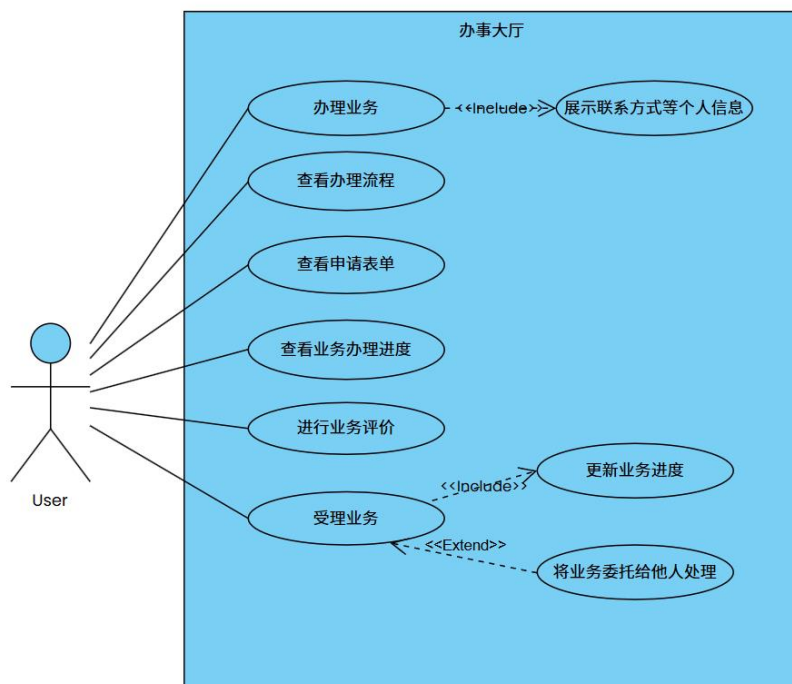


图 2-2 办事大厅用例图（办事大厅层级）

### (2) 详细描述

- a) 用例名：办事大厅登录；
- b) 用例标识：UC-Office；
- c) 主要执行者：用户（学生，教职工）；
- d) 目标：通过合法身份登录门户系统、跳转进入办事大厅后以获得不同的操作权限，如办理业务和受理业务；
- e) 范围：天津大学融合门户系统中的办事大厅；
- f) 前置条件：使用融合门户系统的办事大厅业务板块之时；
- g) 交互动作：用户输入学工号和密码；系统验证用户账号和密码的正确性和合法性，并根据用户的职务分配其权限；验证正确则意味着登录成功。

## 3. 第三层级——某一服务模块

### (1) 奖励学金证明申请服务

#### a) 用例图

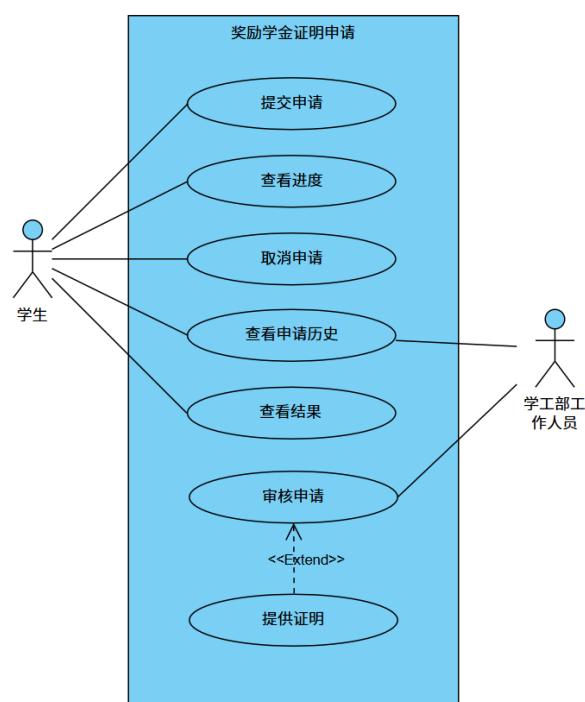


图 2-3 奖励学金证明申请服务用例图（具体服务层级）

b) 详细描述：

- ①用例名：奖励学金申请服务；
- ②用例标识：UC-ScholarshipProof；
- ③主要执行者：用户（学生，学工部工作人员）；
- ④目标：通过合法身份登录门户系统、跳转进入办事大厅后以获得不同的操作权限，如提交奖励学金证明申请或对相关申请进行审核；
- ⑤范围：天津大学融合门户系统中办事大厅板块中的奖励学金申请服务；
- ⑥前置条件：使用融合门户系统的办事大厅业务板块的奖励学金申请服务之时；
- ⑦交互动作：用户输入学工号和密码；系统验证用户账号和密码的正确性和合法性，并根据用户的职务分配其权限；验证正确则意味着登录成功；接着可在合法权限范围内进行操作。

(2) 社团活动审批服务

a) 用例图



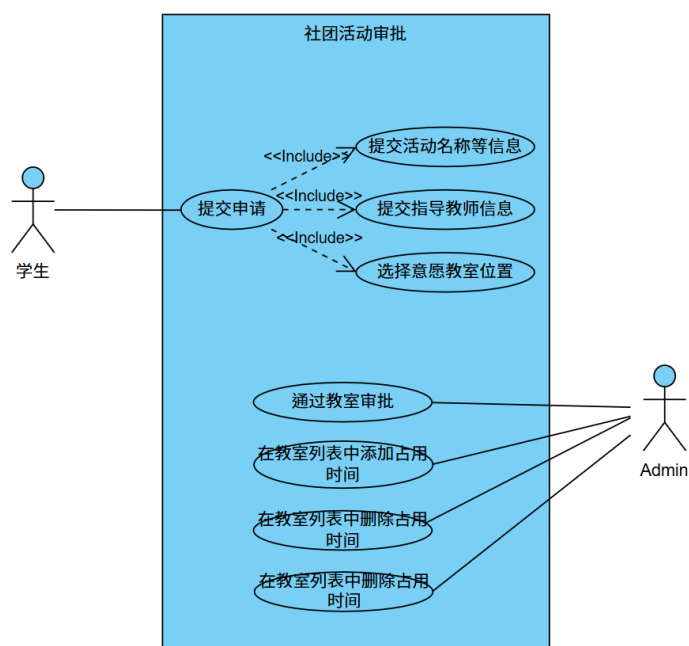


图 2-4 社团活动审批服务用例图（具体服务层级）

b) 详细描述：

- ①用例名：社团活动审批服务；
- ②用例标识：UC-Activities；
- ③主要执行者：用户（学生，学工部工作人员）；
- ④目标：通过合法身份登录门户系统、跳转进入办事大厅后以获得不同的操作权限，如作为学生提交社团活动审批申请以请求在某一时间范围内使用某教室或作为教职工对相关申请进行审批；
- ⑤范围：天津大学融合门户系统中办事大厅板块中的社团活动审批服务；
- ⑥前置条件：使用融合门户系统的办事大厅业务板块的社团活动审批服务之时；
- ⑦交互动作：用户输入学工号和密码；系统验证用户账号和密码的正确性和合法性，并根据用户的职务分配其权限；验证正确则意味着登录成功；接着可在合法权限范围内进行操作。

## 2.1.4 非功能需求

### 1. 系统性能与可靠性

考虑到天津大学每年本科新生约 5000 人，所以本人推断在性能测试中，系统应支持 10000 名并发用户，保证高峰时期响应速度不超过 3 秒。在系统负载较

高时仍应确保稳定的性能表现，保证用户流畅的使用体验。系统需保证高性能运行，同时确保在各种故障情况下的可靠性。

## 2. 安全性

系统必须具备高水平的安全性，所有敏感数据（如用户密码、文件数据）在传输过程中应通过 HTTPS 加密，防止信息泄露。此外，系统需进行定期安全审查，防止 SQL 注入、XSS 等常见安全漏洞。

## 3. 可用性

融合门户系统要求全年 365 天不间断运行，确保师生能够随时访问。系统的可用性应达到 99.9%，即每年不可用时间不超过 8.76 小时。

## 4. 可维护性与可扩展性

系统应当有良好的文档和代码结构并具有良好的可扩充性，确保后期可轻松进行维护和升级。

## 5. 操作系统适配性

系统应能够适配主流的操作系统，如 Windows、Linux、macOS 等。

# 2.2 软件需求优先级及开发安排

为了确保项目的顺利推进和交付，系统开发将按照优先级分为如下三个阶段进行：

### 1. 第一阶段（核心功能开发）

- (1) 优先级：最高。身份验证是系统的基础功能，办事大厅则是用户日常使用频率最高的业务板块。
- (2) 开发安排：开发系统的用户身份验证功能以及办理业务、文件流转、受理业务功能。这两个模块是系统的基础，需在开发初期完成。

### 2. 第二阶段（用户角色功能开发）

- (1) 优先级：中等。角色功能是系统主要业务逻辑所在，应确保在第一阶段核心功能完成后快速开发。
- (2) 开发安排：根据不同用户角色（学生、教职工）的需求，开发相应的功能模块，如融合门户中的办事大厅的电类维修申请服务：学生可提交需维修的电类器件及其所在位置、维修师傅可进行接单并上传维修进度等。

### 3. 第三阶段（系统管理功能开发）

- (1) 优先级：低。虽然系统管理功能对系统的维护至关重要，但相比用户功能，它的使用频率较低，因此安排在开发后期。

- (2) 开发安排：实现管理员后台功能，确保管理员可以维护用户权限、管理公告和备份系统数据。

## 2.3 用例交互模型

下面，我们分别从融合门户系统、办事大厅板块、具体服务这 3 个层级进行阐述，按层次展示系统从整体到具体功能的逐级交互流程。

### 1. 融合门户系统层级

在系统的顶层层级，展示用户如何登录融合门户系统。

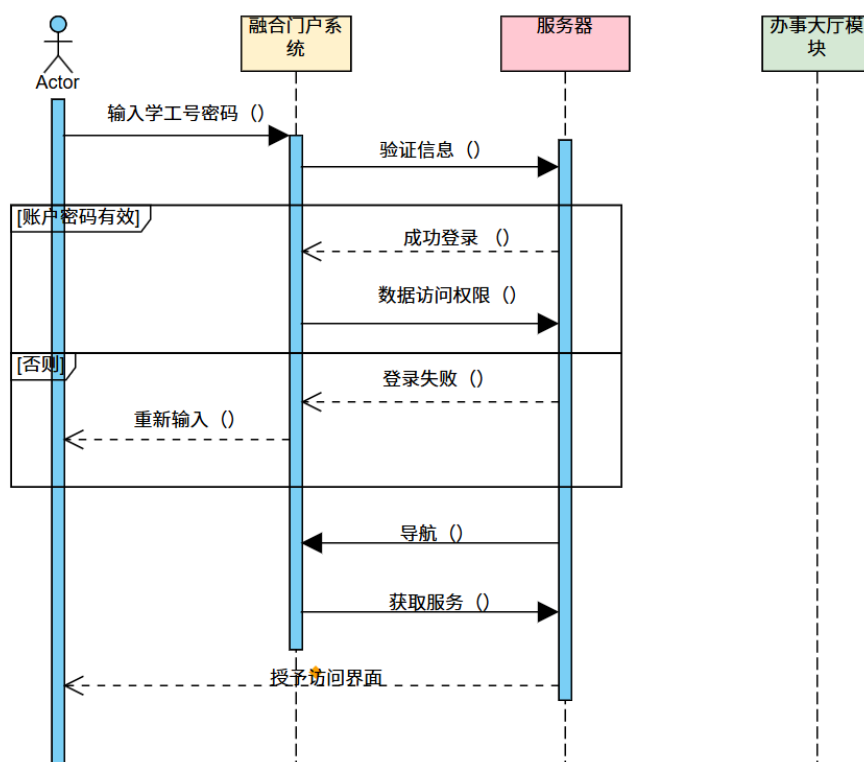


图 2-5 用户登录和服务访问顺序图

对象描述：

- (1) 参与者：用户。用户登录系统并请求访问办事大厅模块。
- (2) 融合门户系统：负责身份验证并将用户导航至指定模块。
- (3) 服务器：验证用户身份并负责服务的导航。
- (4) 办事大厅：一旦验证成功，门户系统为用户提供访问办事大厅的权限。

### 2. 办事大厅模块层级

在办事大厅模块中，用户可以选择不同的服务项目，例如社团活动审批、本科生补办学生证等服务。此顺序图描述用户如何作为普通用户办理业务以及如何作为某项服务的管理者来受理业务。

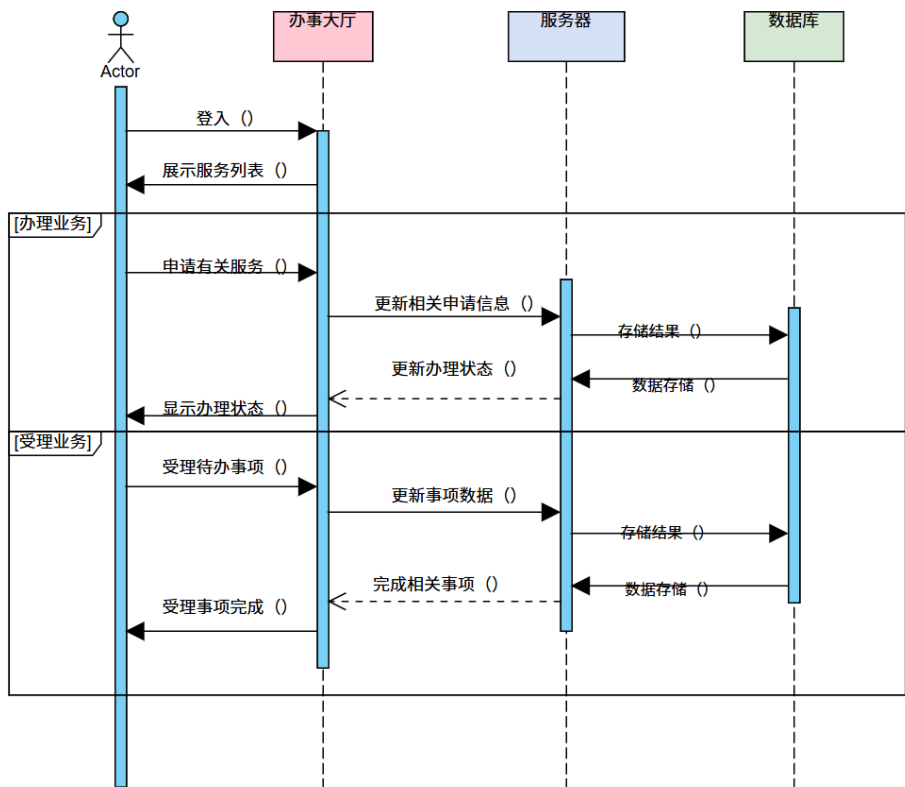


图 2-6 用户办理服务或受理服务顺序图

描述：

- (1) 用户：进入办事大厅后，选择业务进行办理或者查看待办事项并对其进行受理或转交他人受理。
- (2) 办事大厅：展示可选服务，并将用户导航至具体服务。
- (3) 服务器：帮助用户浏览和选择特定服务。
- (4) 数据库：存储业务的办理状态等信息。

3. 办事大厅中的社团活动审批服务层级

以社团活动审批这一具体服务为例。在社团活动审批服务层级，用户提交活动审批申请，系统处理申请并返回结果。

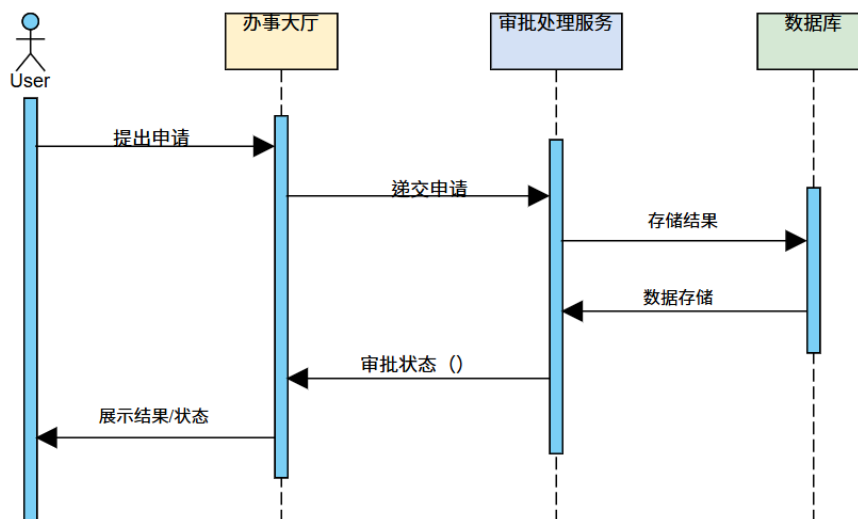


图 2-7 提交审批申请顺序图

描述：

- (1) 用户：提交社团活动审批申请。
- (2) 办事大厅：接收用户请求并调用审批处理服务。
- (3) 审批处理服务：存储审批请求并处理数据。
- (4) 数据库：存储用户的审批申请数据并支持审批流程。

## 2.4 分析类模型

分析类模型主要用于展示系统中不同类之间的关系和结构，以下是“办事大厅”模块中三个具体服务（奖学金证明申请、社团活动审批、门窗类维修申请）的分析类模型。这些模型定义了每个服务的关键类及其属性、方法和相互关系，以明确服务的构成与逻辑结构。

### 1. 奖学金证明申请

#### (1) 主要类：

##### a) User（用户）

属性：userID（学工号）、password（密码）、role（角色：即教职工、学生、管理员）

方法：登录 login()、提交申请 submitRequest()

##### b) ScholarshipApplication（奖学金证明申请）

属性：applicationID（申请记录的 ID）、userID（申请者）、scholarshipType（奖励学金种类）、requestDate（申请时间）、status（状态）

方法：创建申请 `createApplication()`、查看状态 `checkStatus()`、取消申请 `cancelApplication()`

c) `ApprovalOfficer`（审批人员）

属性：`officerID`（审批人员 ID）、`name`（审批人员姓名）、`department`（审批人员所在部门）

方法：查看申请 `reviewApplication()`、同意申请 `approveApplication()`、拒绝申请 `rejectApplication()`

d) `Notification`（通知）

属性：`notificationID`、`userID`、`message`、`date`

方法：`sendNotification()`、`viewNotification()`

(2) 关系：

a) User 提交 `ScholarshipApplication`

b) `ApprovalOfficer` 审批 `ScholarshipApplication`

c) `Notification` 用于向 User 发送结果通知

2. 社团活动审批

(1) 主要类：

a) `User`（用户）

属性：`userID`、`name`、`PhoneNumber`、`role`

方法：`login()`、`submitRequest()`

b) `ActivityApplication`（社团活动审批申请）

属性：`applicationID`、`userID`、`activityName`、`activityDate`、`location`、`status`

方法：`createApplication()`、`checkStatus()`、`updateApplication()`、`cancelApplication()`

c) `ApprovalCommittee`（审批人员）

属性：`committeeID`、`name`、`department`

方法：`evaluateApplication()`、`approveApplication()`、`requestRevisions()`

d) `Document`（文件）

属性：`documentID`、`applicationID`、`filePath`、`uploadDate`

方法：`uploadDocument()`、`downloadDocument()`

(2) 关系：

a) User 创建并更新 `ActivityApplication`

b) `ApprovalCommittee` 审批 `ActivityApplication`

c) `Document` 存储活动相关文件

### 3. 门窗类维修申请

#### (1) 主要类：

##### a) User（用户）

属性：userID、name、PhoneNumber、role

方法：login()、submitRequest()

##### b) RepairRequest（维修申请）

属性：requestID、userID、repairType、location、description、requestDate、status

方法：createRequest()、trackRequestStatus()、cancelRequest()

##### c) MaintenanceWorker（维修人员）

属性：workerID、name、specialization、contactInfo

方法：assignTask()、completeTask()、updateStatus()

##### d) WorkOrder（工单）

属性：workOrderID、requestID、assignedWorker、startDate、endDate、status

方法：generateWorkOrder()、updateWorkOrderStatus()

#### (2) 关系：

a) User 创建 RepairRequest

b) MaintenanceWorker 执行 RepairRequest

c) WorkOrder 跟踪 RepairRequest 的进展

## 2.5 状态模型

下面通过状态图描述系统中关键模块的状态转换过程：

### 1. 针对总体层级

针对总体层级，分析天津大学融合门户的用户类对象（学生以及教职工）的状态图。

状态变化包括：未登录、登录中、已登录、登录失败。用户进入系统时首先处于“未登录”状态，输入凭据后进入“登录中”状态，验证成功则进入“已登录”状态，验证失败则进入“登录失败”状态，并返回登录界面重新登录（重新进行上述状态转换）。状态图如下：

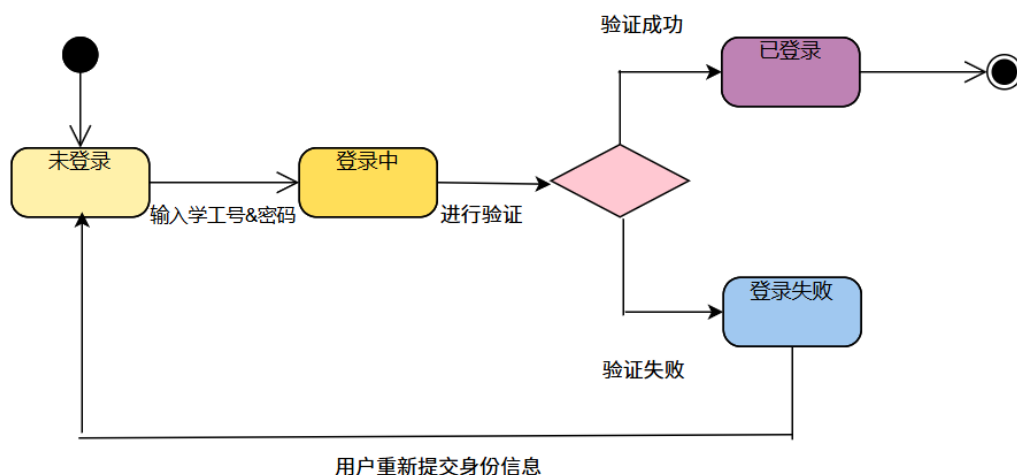


图 2-8 用户身份验证状态图

## 2. 针对办事大厅模块层级

针对办事大厅模块层级，分析融合门户系统的办事大厅模块的用户类对象（负责办理业务和受理业务）的状态图。

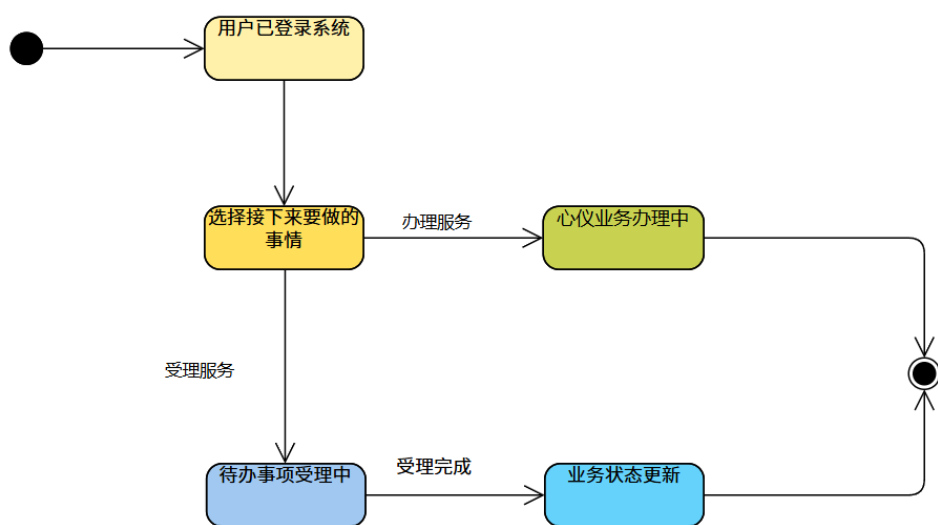


图 2-9 业务处理状态图

状态变化包括：用户已登录系统、心仪业务办理中、待办事项受理中、业务状态更新。用户登录系统后，状态变更为“用户已登陆系统”；接着选择要做的事情：若进行办理服务，则状态将变更为“心仪业务办理中”，随后达到终态；若进行受理业务，则状态将变为“待办事项受理中”，受理完成后，状态变更为“业务状态更新”，随后达到终态。

## 3. 针对办事大厅模块中的某一服务（社团活动审批）这一层级



分析融合门户系统办事大厅模块的社团活动审批服务的用户类对象的状态图。

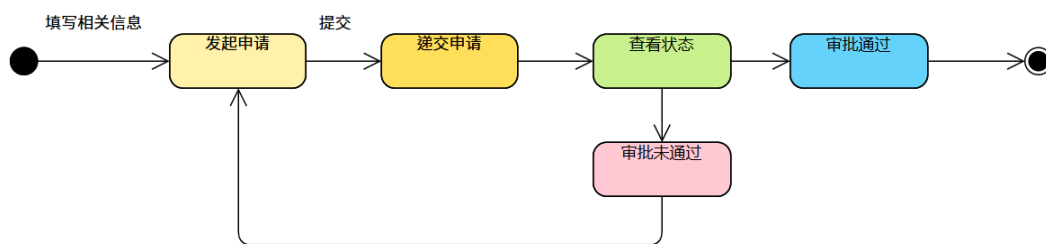


图 2-10 审批处理状态图

状态变化包括：发起申请、递交申请、查看状态或结果。用户填写相关信息后，状态变更为“发起申请”；提交后，状态变更为申请已提交，之后可查询状态或结果，若审批通过，则随后达到终态；若审批未通过，则返回发起申请状态，重新执行后续状态流程。

## 3 体系结构设计

体系结构设计是软件系统设计的重要环节，决定了系统的逻辑组成、功能模块之间的关系以及其部署方式。通过对天津大学融合门户系统的深入分析，采用面向对象的设计思想，结合 UML 建模工具，下面从逻辑视图和部署视图两个层面对系统进行设计。

### 3.1 体系结构的逻辑视图

逻辑视图描述了系统中各个模块的组织结构，展示了系统的主要功能组件及其相互之间的依赖关系。天津大学融合门户系统由多个功能模块组成，各模块之间相互协作，共同完成门户的各项功能。接下来以系统的主要逻辑组成部分逐层展开分析。

#### 3.1.1 包图

系统通过包图展示主要模块及其相互关系，并刻画包间的构成和依赖关系。下面仍然根据融合门户系统、办事大厅、某项具体服务三个层级逐级展开。

##### 1. 针对总体层级——天津大学融合门户系统

下图展示了天津大学融合门户系统的包图：

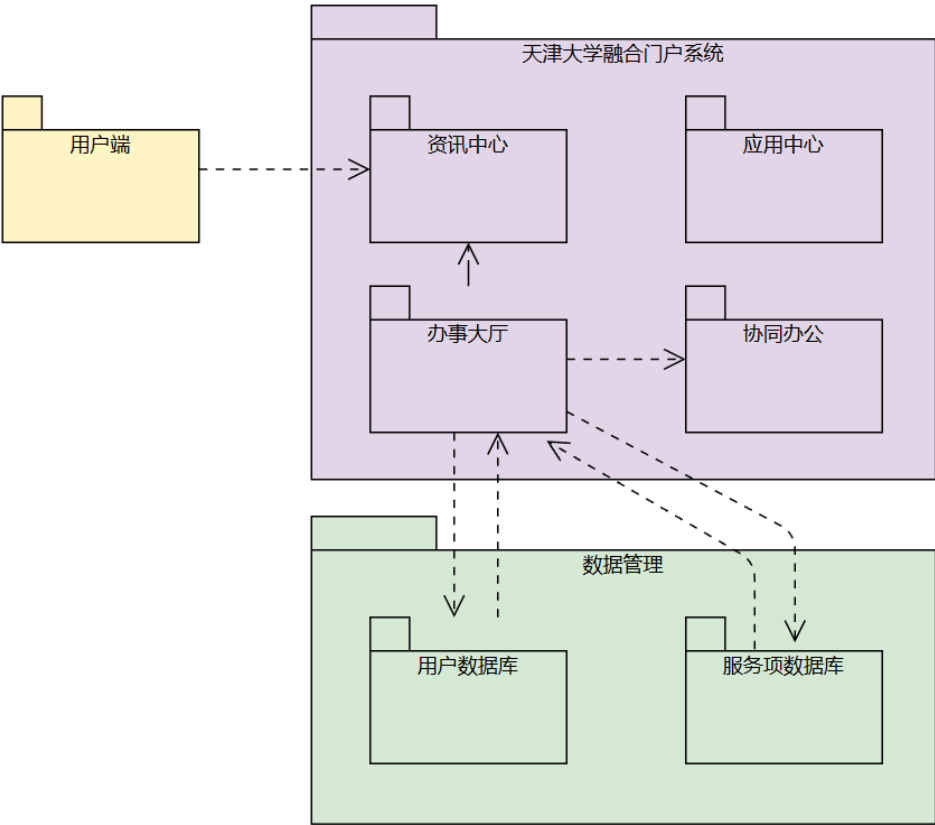


图 3-1 天津大学融合门户系统的包图（总体层级）

其中，门户系统模块包含 4 个功能模块，分别是资讯中心、办事大厅、应用中心、协同办公。数据管理模块用于提供用户数据库、服务项数据库查询，可查询用户申请或审批的服务，以及用户查询的某项服务的办理流程及进度。

2. 针对办事大厅模块层级

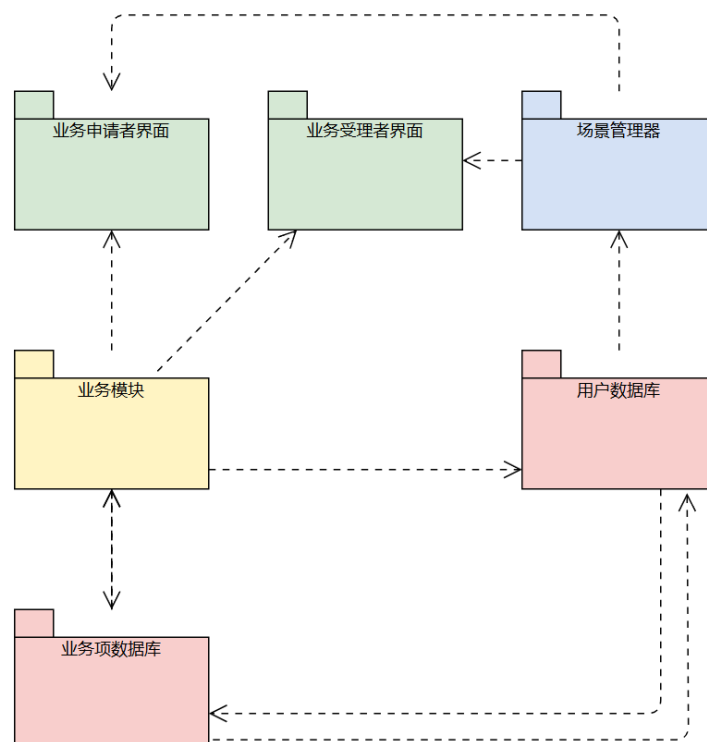


图 3-2 天津大学融合门户系统办事大厅的包图（办事大厅层级）

本图中包间关系均为依赖关系。业务申请者界面和业务受理者界面需要来自业务模块提供的服务，同时，业务模块和业务项数据库间存在双向依赖：业务模块向业务项数据库查询服务办理进度，业务项数据库返回相关信息。场景管理器向业务受理者界面提供业务办理进度等信息，同时接收用户数据库的信息，形成包含服务相关的所有信息（如用户、申请服务名称、时间等）完整数据链，向用户提供正确、完整的信息。

### 3. 针对办事大厅模块中的某一服务（社团活动审批）这一层级

包图如下图所示：

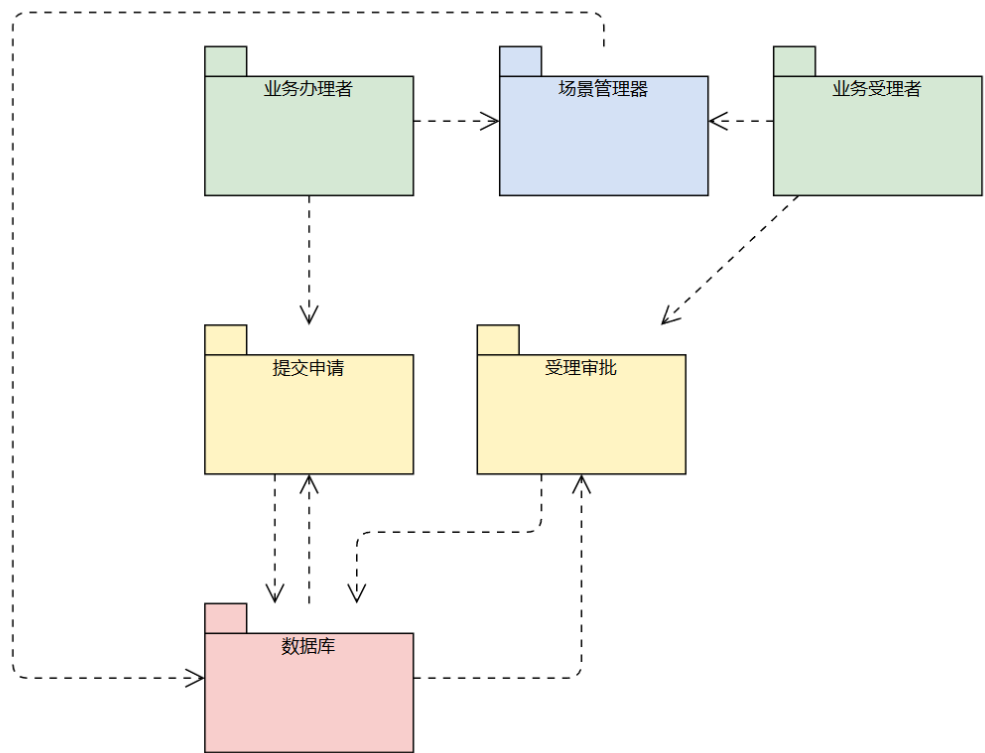


图 3-3 社团活动审批的包图（具体服务层级）

本图中包间关系均为依赖关系。业务申请者界面依赖于提交申请模块所提供的服务，而业务受理者界面需要来自受理审批模块提供的服务。同时，场景管理器向业务申请者界面和业务受理者界面提供业务办理进度等信息，并依赖于数据库模块，向其查询办理流程进度：二者分别向数据库请求当前业务状态，并将更新后的状态信息传递给数据库。

### 3.1.2 构件图

构件图展示了系统的实现组件及其相互交互关系。下面仍然根据融合门户系统、办事大厅、某项具体服务三个层级逐级展开描述对应部分的构件图。

#### 1. 针对总体层级——天津大学融合门户系统

天津大学融合门户系统中，各功能模块通过以下构件进行实现：

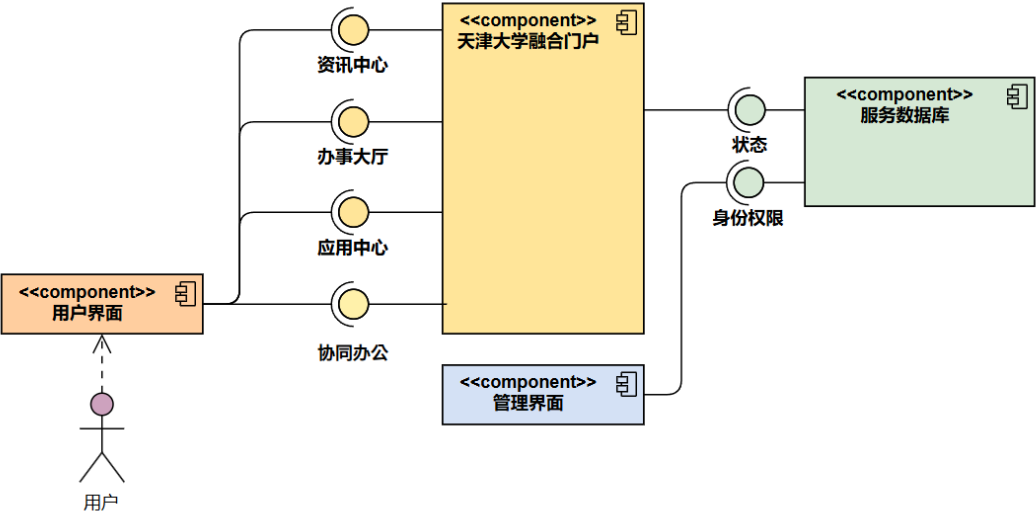


图 3-4 天津大学融合门户系统的构件图（总体层级）

融合门户对外提供资讯中心、办事大厅、应用中心、协同办公 4 个模块接口，用户界面使用来自天津大学融合门户的 4 个接口，服务数据库对外提供状态、身份权限这 2 个接口，管理界面应用来自服务数据库提供的身份权限接口，以对普通用户、额外具有某项服务受理权限的用户、管理员提供不同的业务权限。融合门户系统使用服务数据库提供的状态接口，以应对用户对某项服务的办理状态查询操作。

2. 针对办事大厅模块层级

构件图如下：

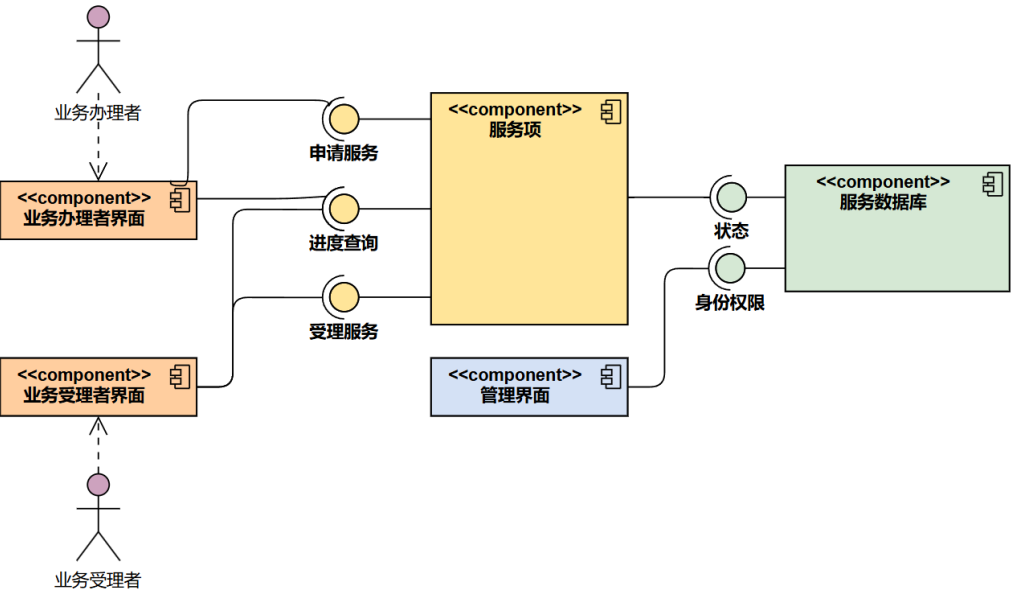


图 3-5 门户系统办事大厅模块的构件图（办事大厅层级）

服务项对外提供申请服务、进度查询、受理服务 3 个模块接口；业务办理者界面使用来自服务项的 2 个接口，分别为申请服务、进度查询；业务受理者界面使用来自服务项的 2 个接口，分别为进度查询、受理服务；服务数据库对外提供状态、身份权限这 2 个接口，管理界面应用来自服务数据库提供的身份权限接口，以对普通用户、额外具有某项服务受理权限的用户、管理员提供不同的业务权限。融合门户系统使用服务数据库提供的状态接口，以应对用户对某项服务的办理状态查询操作。

3. 针对办事大厅模块中的某一服务（以社团活动审批为例）这一层级  
构件图如下：

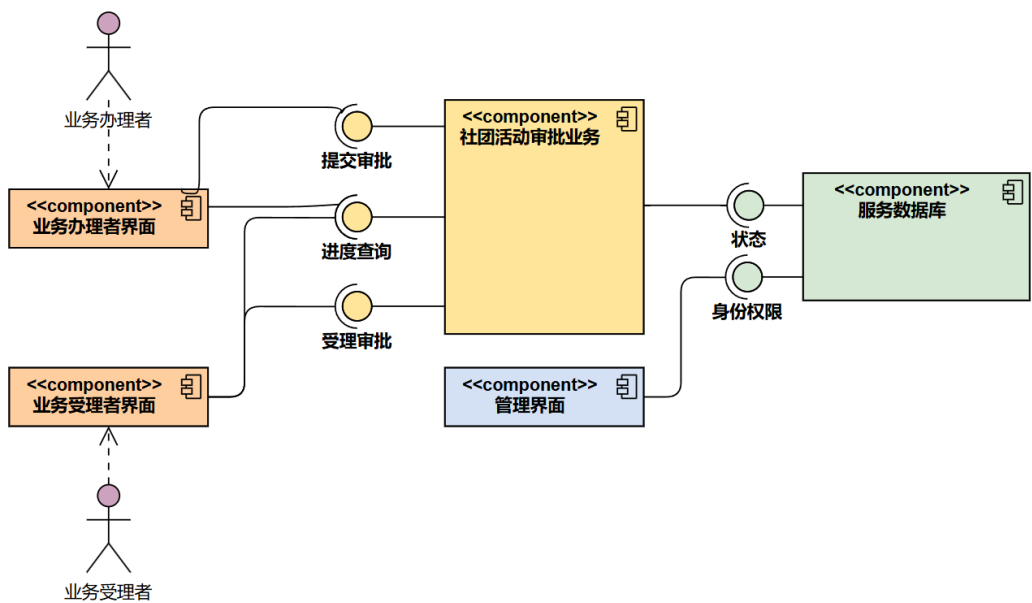


图 3-6 门户系统的社团活动审批服务构件图（具体服务层级）

社团活动审批业务对外提供提交审批、进度查询、受理审批 3 个模块接口；业务办理者界面使用来自社团活动审批业务的 2 个接口，分别为提交审批、进度查询；业务受理者界面使用来自社团活动审批业务的 2 个接口，分别为进度查询、受理审批；服务数据库对外提供状态、身份权限这 2 个接口，管理界面应用来自服务数据库提供的身份权限接口，以对普通用户、额外具有某项服务受理权限的用户、管理员提供不同的业务权限，确保系统安全性。融合门户系统使用服务数据库提供的状态接口，以应对用户对某项服务的办理状态查询操作。

### 3.2 体系结构的部署视图

部署视图展示了系统在物理硬件上的部署方式，天津大学融合门户系统部署在一个分布式的网络环境中，通过服务集群和数据库服务器支撑系统的运行。

### 3.2.1 部署架构描述

天津大学融合门户系统的部署环境包含以下主要组件：

1. **Web 服务器：**负责接收用户请求，处理前端展示和接口调用。
2. **应用服务器：**负责处理业务逻辑，如用户管理、服务办理及受理、资讯发布与浏览等。
3. **数据库服务器：**存储用户数据、服务办理数据、资讯信息以及其他相关信息。

上述服务器共同构成系统的核心运行环境，采用多层架构来保证系统的可扩展性和性能。

### 3.2.2 部署图

部署图如下所示，展示了天津大学融合门户系统的各个物理节点和它们之间的通信关系：

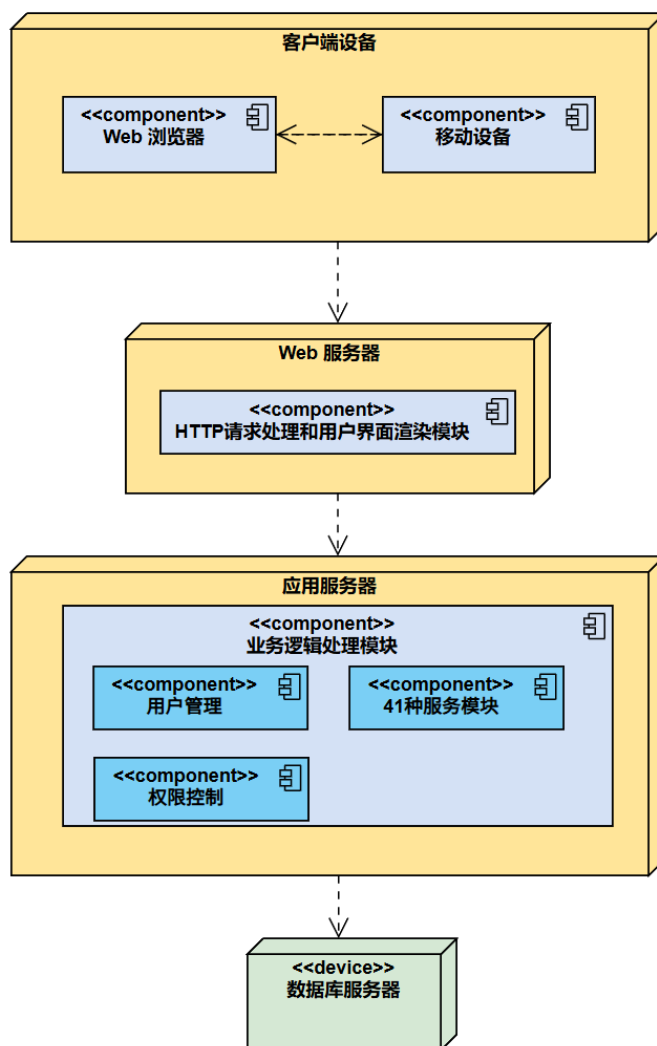


图 3-7 天津大学融合门户系统部署图

### 3.2.3 关键技术及安全策略

查询资料总结得出软件体系结构设计的关键技术及安全策略如下：

1. 分布式架构：系统采用分布式部署方式，通过负载均衡提高系统的性能和稳定性。
2. 数据安全：所有敏感数据在传输过程中可通过 SSL 加密，同时在数据库中使用权限控制机制，确保只有授权用户能够访问相关数据。
3. 备份与恢复：数据库部署备份策略，确保在发生系统故障时能够迅速恢复数据，保证系统的高可用性。



## 4 详细设计

详细设计部分基于需求分析的结果，进一步明确系统内部各个模块的实现方式。下文将展示分析和建立软件需求模型的具体步骤，通过用例设计、类设计、数据设计和活动图，逐步描述天津大学融合门户系统的具体实现过程。

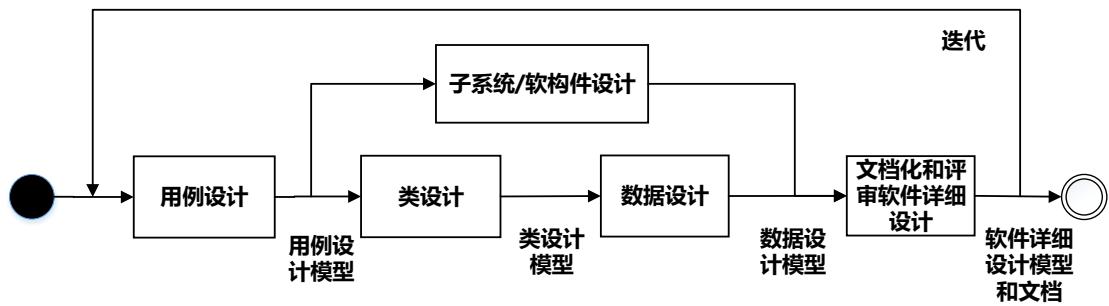


图 4-1 详细设计状态图

### 4.1 用例设计

详细分析和建立软件需求模型的第一步骤是分析和建立用例交互模型。在这一步骤中，我们首先要分析和确定用例所涉及的对象类；接着分析和确定对象间的消息传递；最后绘制用例的交互图。用例设计展示了系统中主要用例的具体实现流程，下面通过顺序图描述各用例之间的交互过程。

#### 4.1.1 用例 1：用户登录

在融合门户系统层级，展示用户如何登录融合门户系统，用例实现方案的顺序图如下。

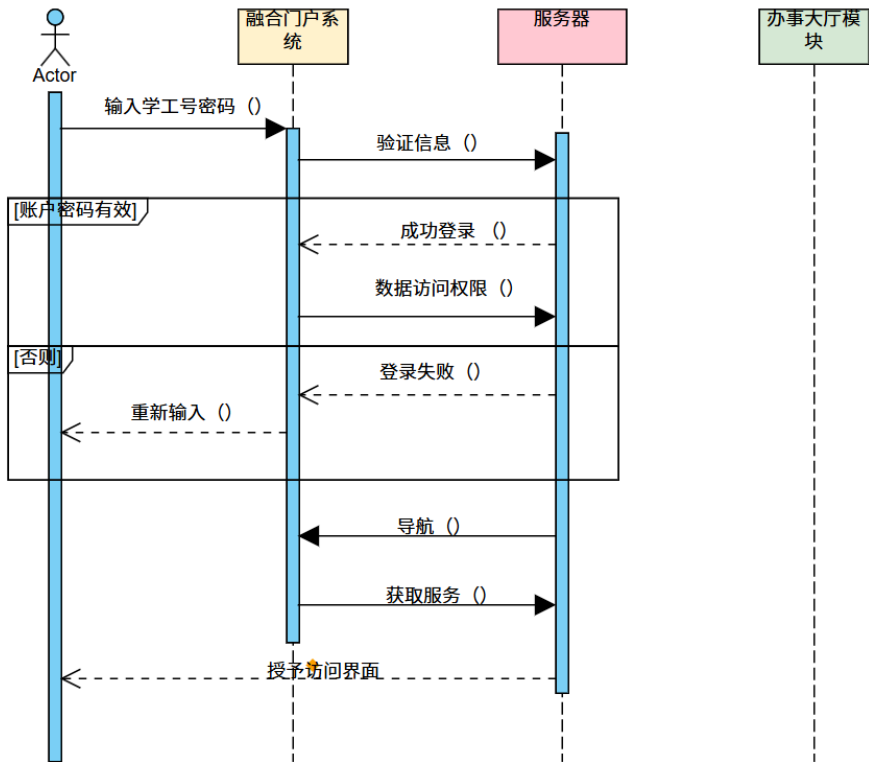


图 4-2 用户登录和服务访问顺序图

相关描述：

- (1) 参与者：用户。用户登录系统并请求访问办事大厅模块。
- (2) 前置条件：用户已在系统中注册，具备登录权限
- (3) 融合门户系统：负责身份验证并将用户导航至指定模块。
- (4) 服务器：验证用户身份并负责服务的导航。
- (5) 办事大厅：一旦验证成功，门户系统为用户提供访问办事大厅的权限。

这里用户首先输入学工号及密码，该消息传递到融合门户系统这一边界类中，接着融合门户系统向服务器发送验证信息请求，若此时账户密码有效，则服务器返回信息，通知门户系统该用户可成功登录，融合门户系统随后向服务器请求数据访问权限；若账号密码无效，则服务器向门户系统返回登录失败的消息，门户系统向用户返回信息，通知其重新输入。此后，在用户登录成功、门户系统向数据库请求数据访问权限的基础上，服务器向门户系统发送导航指引，门户系统向服务器发送获取信息调用。最后，服务器向用户返回授权访问的界面，用户进而可以进行办理业务等后续操作。

4.1.2 用例 2：用户与办事大厅的交互

在融合门户系统中办理社团活动审批业务中，展示用户如何办理业务并

对已办理的服务进行进度查询，用例实现方案的顺序图如下。

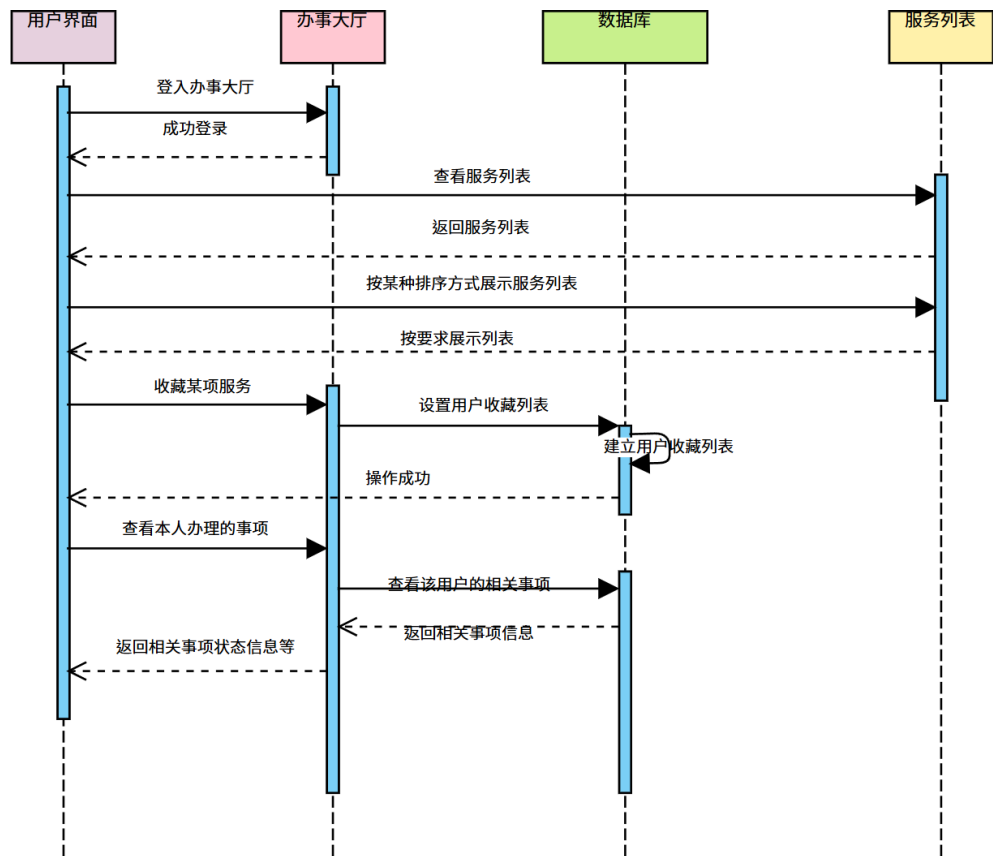


图 4-3 用户与办事大厅的交互顺序图

相关描述：

- (1) 参与者：尝试登入办事大厅查询业务相关信息的用户。用户在办事大厅中可进行按默认排序、拼音排序、申请数目 3 种排序方式查询开放的服务；并可对某项服务进行收藏；同时，用户可以查看自己办理的事项，包括已办结和正在办理两种状态。
- (2) 前置条件：用户已登录天津大学融合门户系统。
- (3) 办事大厅：作为边界类作为用户查询与收藏服务功能的一部分。
- (4) 服务列表：作为实体类，存储 41 种服务类型，接受用户的查询列表调用操作，并返回按用户需求排序的服务列表。
- (5) 数据库：存储用户、用户办理的服务、用户收藏的服务列表等信息，与具体服务和用户、办事大厅等模块进行交互。

用户在成功登入天津大学融合门户系统后通过用户界面进入办事大厅，随后办事大厅向用户界面返回成功登录指示，随后用户通过用户界面查询服务列表，服务列表模块随即向用户返回服务列表；当然，用户也可在用户界面上进行服务列表排序操作（包括默认升降排序、按拼音升降排序、按申请

数升降排序几种方式)，服务模块随后向用户返回其要求的排序方式的列表，便于用户筛选查找服务。用户还可以在界面对某项服务进行收藏操作，该信息将依次传至办事大厅模块、数据库；随后数据库进行自调用，建立用户收藏列表，并将操作成功消息传回用户界面。此外，用户还可在用户界面顶端进行查找本人办理的事项，该信息调用依次传递至办事大厅、数据库，随后数据库将相关该用户的事项信息返回办事大厅，办事大厅再将该信息传回至用户界面，至此，除办理业务事件外，所有的交互动作已描述完毕。

### 4.1.3 用例 3：社团活动审批业务办理

在融合门户系统中办理社团活动审批业务中，展示用户如何办理业务并对已办理的服务进行进度查询，用例实现方案的顺序图如下。

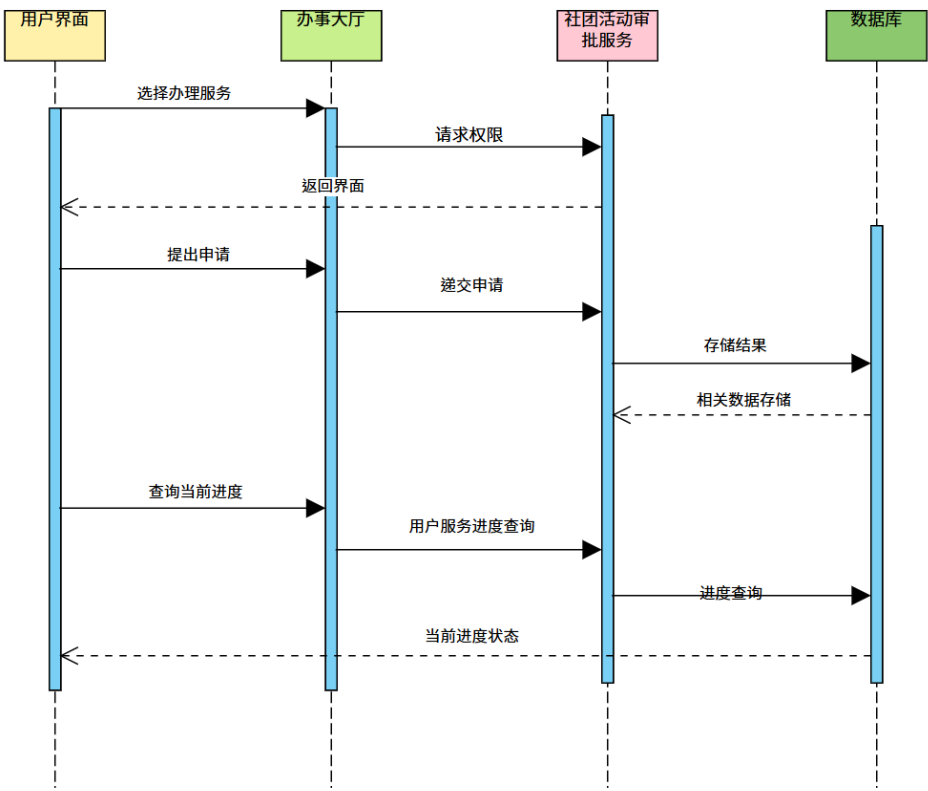


图 4-4 社团活动审批业务办理的交互顺序图

对象描述：

- (1) 参与者：尝试办理业务的用户。用户在办事大厅中选择想要办理的服务。
- (2) 前置条件：用户已登录天津大学融合门户系统。
- (3) 办事大厅：作为边界类作为用户提交申请、进度查询功能的一部分。

- (4) 社团活动审批服务模块：向数据库发送数据存储及查询调用，并接收来自办事大厅的服务相关信息。
- (5) 数据库：存储用户、用户办理的服务、服务进度时间及流转状态等信息，与具体服务和用户进行交互。

用户通过用户界面选择办事大厅中的某项业务（本例为社团活动审批）进行办理，办事大厅向社团活动审批模块请求权限，接着社团活动审批模块向用户界面返回界面，支持用户进行后续申请操作。接着用户提出申请到办事大厅，办事大厅将该用户提交申请的消息传递至社团活动审批模块，此后该服务模块向数据库发送存储结果的消息，存储完成后数据库向服务模块返回相关数据存储成功的消息。此后，用户向办事大厅发送查询当前进度的请求，该查询调用依次通过办事大厅、社团活动审批业务模块、最后向数据库进行查询调用。数据库将查询到的该用户对社团活动审批服务的办理状态返回至用户界面，至此一轮交互动作完成。

#### 4.1.4 设计类图

##### 1. 用户类设计

用户的设计类图如下：

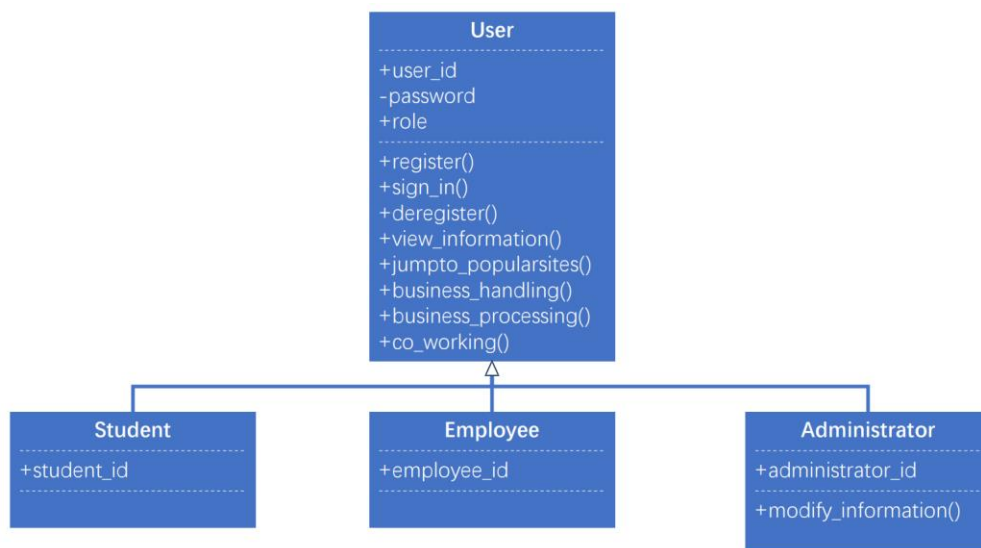


图 4-5 用户类设计类图

首先时 User（用户）类，拥有 user\_id（用户 ID）、password（密码）、role（角色）3 个属性；有注册 register()、登录 sign\_in()、登出 deregister()、查看公告 view\_information()、在应用中心跳转至常用网站 jumpto\_popularsites()、办理业务 business\_handling()、受理业务 business\_processing()、协同办公 co\_working()这 8 个功能方法。

此外，用户类分为学生 Student、教职工 Employee、管理员 Administrator 三种类型，三者分别特有 `student_id`、`employee_id`、`administrator_id` 属性，另外管理员类还特有修改公告信息 `modify_information()` 功能。三者与用户类为继承关系。

## 2. 服务类设计

服务类的设计类图如下：

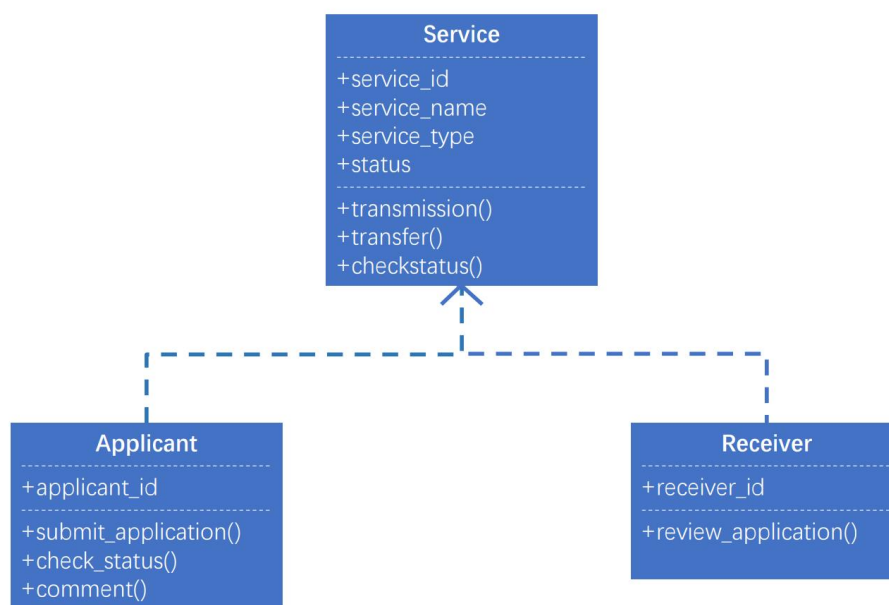


图 4-6 服务类设计类图

首先是 Service 类，包含服务项 ID 号 `service_id`、服务名称 `service_name`、服务种类 `service_type`、服务状态 `status` 这 4 个属性；拥有申请流转 `transmission()`、状态变更 `transfer()`、查看状态 `checkstatus()` 这 3 个方法。

接着是 Applicant 类，包含 `applicant_id`（申请者 ID）这 1 个属性；拥有递交申请 `submit_application`、查看办理状态 `check_status()`、评价 `comment()` 这 3 个方法。

最后是 Receiver 类，包含 `receiver_id` 这个属性，拥有受理申请 `review_application()` 这一方法。

## 4.2 类设计

类设计部分通过类图详细描述系统中各类的属性、方法及其关系，明确系统中各对象的行为与交互。

### 4.2.1 用户类设计

用户类包含系统中所有用户的公共属性和方法,通过继承机制实现不同角色的特定行为。

由图 4-5 的用户设计类图得出以下内容:

1. 可见范围
  - (1) User 类中, 包含 3 个属性, 其中 `user_id` (用户 ID) 和 `role` (角色) 的可见性为 `public`, `password` (密码) 的可见性为 `private`。
  - (2) Student 类中, 在继承 User 类的三个属性的基础上, 特有一个属性 `student_id`, 可见性为 `public`。
  - (3) Employee 类中, 在继承 User 类的三个属性的基础上, 特有一个属性 `employee_id`, 可见性为 `public`。
  - (4) Administrator 类中, 在继承 User 类的三个属性的基础上, 特有一个属性 `administrator_id`, 可见性为 `public`。
2. 类间关系

学生 Student、教职工 Employee、管理员 Administrator 类分别和 User 类构成继承关系。
3. 属性设计
  - (1) User 类
    - a) 元素类型  
`user_id` (用户 ID) 和 `password` (密码) 为用户输入元素, `role` (角色) 为静态元素。
    - b) 可见性  
已在前文描述, 这里不再赘述。
    - c) 变量类型与初始值  
`user_id` (用户 ID) 和 `password` (密码) 为字符串变量, 初始值为空串; `role` (角色) 为字符串变量, 初始值为 `student`。
  - (2) 学生 Student 类
    - a) 元素类型  
除从 User 类中继承的属性, 特有属性 `student_id` 为用户输入元素。
    - b) 可见性  
已在前文描述, 这里不再赘述。
    - c) 变量类型与初始值  
特有属性 `student_id` 为字符串变量, 初始值为空串。

(3) 教职工 Employee 类

a) 元素类型

除从 User 类中继承的属性，特有属性 `emplyee_id` 为用户输入元素。

b) 可见性

已在前文描述，这里不再赘述。

c) 变量类型与初始值

特有属性 `emplyee_id` 为字符串变量，初始值为空串。

(4) 管理员 Administrator 类

a) 元素类型

除从 User 类中继承的属性，特有属性 `administrator_id` 为用户输入元素。

b) 可见性

已在前文描述，这里不再赘述。

c) 变量类型与初始值

特有属性 `administrator_id` 为字符串变量，初始值为空串。

4. 方法设计

(1) User 类

有注册 `register()`、登录 `sign_in()`、登出 `deregister()`、查看公告 `view_information()`、在应用中心跳转至常用网站 `jump_to_popularsites()`、办理业务 `business_handling()`、受理业务 `business_processing()`、协同办公 `co_working()` 这 8 个方法。

(2) Administrator 类

特有 `modify_information()` 方法。

5. `business_handling()` 方法实现活动图

(1) 办理社团活动审批业务的活动图



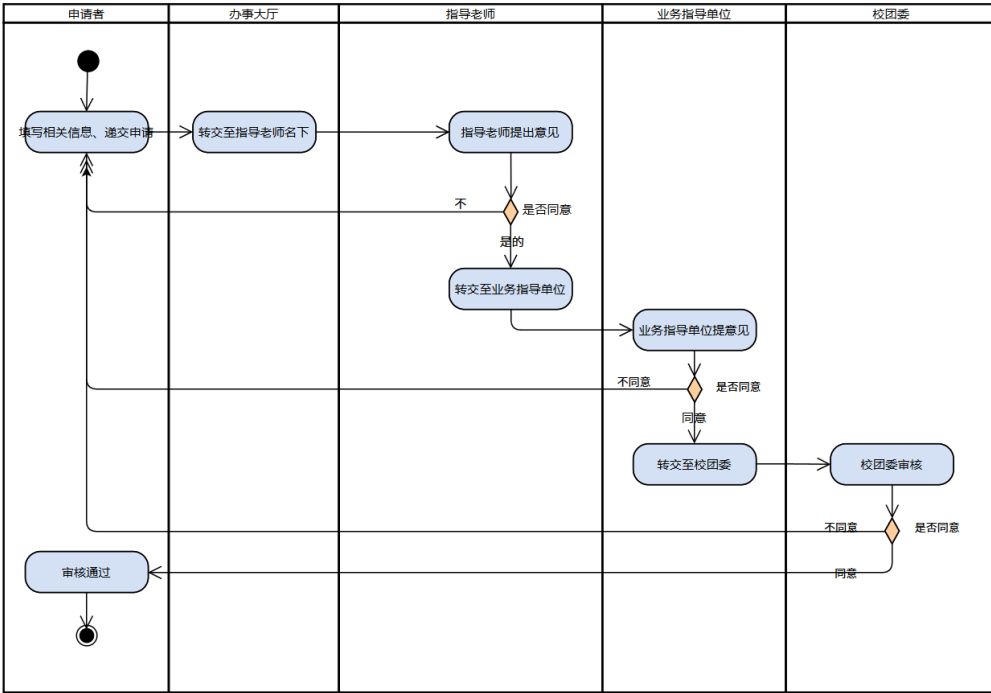


图 4-7 办理社团活动审批业务的活动图

(2) 办理社团活动审批业务的活动图

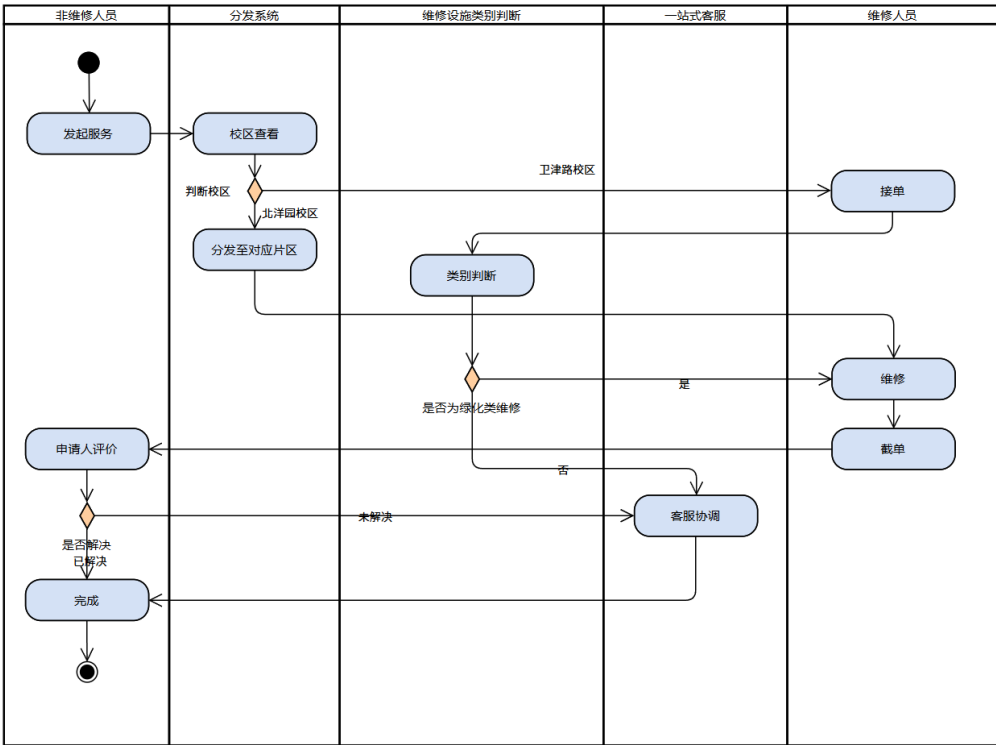


图 4-8 办理绿化类维修申请业务的活动图

## 4.2.2 服务类设计

课程类负责管理系统中的课程信息，包含课程基本信息及与用户的交互方法。

由图 4-6 的服务类设计类图得出以下内容：

### 1. 可见范围

- (1) Service 类中，包含 4 个属性，其中所有属性的可见性为 public。
- (2) Applicant 类中，applicant\_id 属性的可见性为 public。
- (3) Receiver 类中，receiver\_id 属性的可见性为 public。

### 2. 类间关系

Applicant 类、Receiver 类分别和 Service 类构成依赖关系。

### 3. 属性设计

#### (1) Service 类

##### a) 元素类型

服务项 ID 号 service\_id、服务名称 service\_name、服务种类 service\_type、服务状态 status 均为静态元素。

##### b) 可见性

已在前文描述，这里不再赘述。

##### c) 变量类型与初始值

service\_id、service\_name、service\_type、status 均为字符串变量，初始值为空。

#### (2) Applicant 类

##### a) 元素类型

applicant\_id 为用户输入元素。

##### b) 可见性

已在前文描述，这里不再赘述。

##### c) 变量类型与初始值

applicant\_id 为字符串变量，初始值为空串。

#### (3) Receiver 类

##### a) 元素类型

receiver\_id 为静态元素。

##### b) 可见性

已在前文描述，这里不再赘述。

##### c) 变量类型与初始值

receiver\_id 为字符串变量，初始值为空串。

#### 4. 方法设计

##### (1) Service 类

拥有申请流转 `transmission()`、状态变更 `transfer()`、查看状态 `checkstatus()` 这 3 个方法。

##### (2) Applicant 类

拥有递交申请 `submit_application`、查看办理状态 `check_status()`、评价 `comment()` 这 3 个方法。

##### (3) Receiver 类

拥有受理申请 `review_application()` 这一方法。

### 4.3 数据设计

数据设计部分描述系统中的数据存储结构，定义了数据库的表结构及其关系。

#### 4.3.1 数据库表设计

融合门户系统的办事大厅板块提供 41 种服务，每种服务的数据库表略有区别，这里主要展示含有常用属性的数据库表。门户系统数据库由多张表组成，主要包括用户表、服务表等。

用户表 (User):

字段：用户 ID、用户名、密码、角色、注册时间

描述：存储所有用户的基本信息

#### 4.3.2 数据库表结构示例

##### 1. 用户数据库表

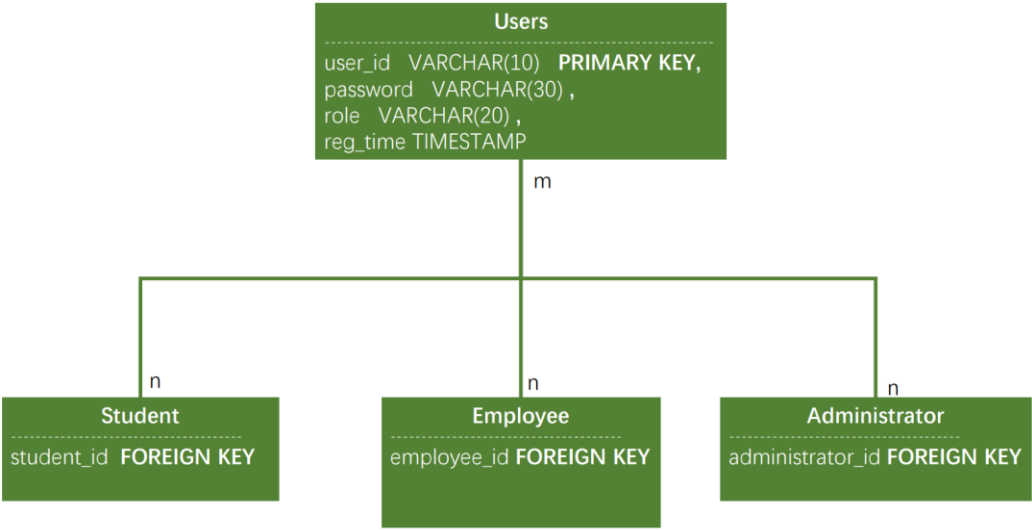


图 4-9 用户数据库表

下面通过表格形式对重要属性进行阐述：

表 4-1 用户表相关属性描述

字段	数据类型	是否 为空	键	默认值	描述
user_id	VARCHAR(10)	NO	Primary Key	NULL	用户 ID (学工号)
password	VARCHAR(30)	NO		NULL	密码
role	VARCHAR(20)	NO		NULL	用户角色
reg_time	TIMESTAMP	NO		CURRENT_TIMESTAMP	注册时间

2. 服务办理数据库表

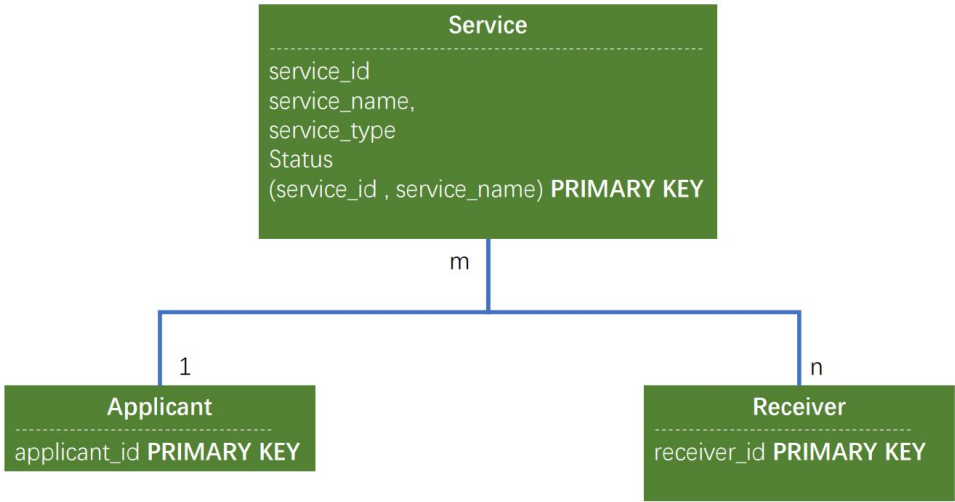


图 4-10 服务办理数据库表

下面通过表格形式对重要属性进行阐述：

表 4-2 业务办理表相关属性描述

字段	数据类型	是否 为 空	键	默认值	描述
applicant_id	VARCHAR(10)	NO		NULL	业务申请者 ID
receiver_id	VARCHAR(10)	NO		NULL	业务受理者 ID
service_id	VARCHAR(30)	NO	Primary Key	NULL	业务单号
service_name	VARCHAR(30)	NO		NULL	业务名称
status	VARCHAR(20)	NO		student	办理状态
time	TIMESTAMP	NO		CURRENT_TIMESTAMP	办结时间

## 4.4 子系统与构件设计

### 4.4.1 系统整体架构

办事大厅模块通过微服务或组件化的方式构建，以便管理多项服务。系统包括用户管理、审批处理、通知和日志四个核心子系统。每个子系统负责不同的任务，协同完成服务的申请与审批流程。

### 4.4.2 子系统设计

#### 1. 用户管理子系统

职责：管理用户身份验证、权限控制和用户信息维护。

##### (1) 构件 1：用户身份验证（Authentication）

a) 功能：验证用户登录信息，确保用户有权限访问办事大厅模块。

b) 方法：

`authenticateUser(userID, password)`: 验证用户凭证。

`logout(userID)`: 用户退出登录。

##### (2) 构件 2：用户信息管理（User Profile Management）

a) 功能：存储并维护用户信息。

b) 方法：

`updateUserInfo(userID, newInfo)`: 更新用户信息。

`getUserInfo(userID)`: 获取用户详细信息。

#### 2. 服务申请管理子系统

职责：管理具体服务的申请处理，包括奖学金证明、社团活动审批和维修申请等 41 项具体服务，这里列举 3 种代表性服务进行详细阐述。

##### (1) 构件 1：奖学金证明申请管理（Scholarship Application Management）

a) 功能：创建、查看和取消奖学金证明申请。

b) 方法：

`createScholarshipApplication(userID, details)`: 用户创建奖学金证明申请。

`checkApplicationStatus(applicationID)`: 查询申请状态。

`cancelApplication(applicationID)`: 用户取消申请。

##### (2) 构件 2：社团活动审批管理（Activity Approval Management）

a) 功能：管理社团活动的申请、文件上传和审批流程。

b) 方法：

`createActivityApplication (userID, activityDetails)`：创建社团活动申请。

`uploadActivityDocuments (applicationID, document)`：上传申请所需信息及文件。

`getApprovalStatus (applicationID)`：查看审批状态。

(3) 构件 3：维修类申请管理（**Repair Request Management**）

a) 功能：用户提交门窗维修申请，并追踪处理状态。

b) 方法：

`submitRepairRequest (userID, repairDetails)`：提交维修申请。

`trackRepairStatus (requestID)`：查询维修申请状态。

`cancelRepairRequest (requestID)`：取消维修申请。

3. 审批处理子系统

职责：审批委员会或相关工作人员处理用户提交的申请。

(1) 构件 1：申请审批（**Application Approval**）

a) 功能：审批用户提交的不同类型的申请（奖学金证明、社团活动等）。

b) 方法：

`approveApplication(applicationID, approverID)`：批准申请。

`rejectApplication(applicationID, approverID, reason)`：拒绝申请并提供原因。

`requestAdditionalInfo(applicationID, details)`：请求补充信息。

4. 通知与日志子系统

职责：将审批结果或状态更新通知用户，并记录用户操作日志。

(1) 构件 1：通知服务（**Notification Service**）

a) 功能：发送状态变更和审批结果的通知给用户。

b) 方法：

`sendNotification(userID, message)`：向用户发送通知。

`viewNotification(notificationID)`：用户查看通知内容。

(2) 构件 2：日志记录（**Logging Service**）

a) 功能：记录用户与系统的交互日志，用于审计。

b) 方法：

`logAction(userID, actionDetails)`：记录用户操作。

`getLogs(userID, filters)`：查询特定日志记录。

### 4.4.3 交互流程

每个子系统通过 REST API 或内部方法调用与其他子系统或构件交互。

例如：

1. 奖学金证明申请流程：
  - (1) 用户登录并选择奖学金证明申请
  - (2) 用户管理子系统（用户身份验证）
  - (3) 服务申请管理子系统（奖学金证明申请管理）
  - (4) 审批处理子系统
  - (5) 通知与日志子系统
2. 社团活动审批流程：
  - (1) 用户登录并选择社团活动审批
  - (2) 用户管理子系统（用户身份验证）
  - (3) 服务申请管理子系统（社团活动审批管理）
  - (4) 审批处理子系统
  - (5) 通知与日志子系统
3. 门窗维修申请流程：
  - (1) 用户登录并选择门窗维修申请
  - (2) 用户管理子系统（用户身份验证）
  - (3) 服务申请管理子系统（维修申请管理）
  - (4) 审批处理子系统
  - (5) 通知与日志子系统

### 4.4.4 数据库设计

数据库中包含用户表（Users）、申请表（Applications）、审批记录表（ApprovalRecords）和日志表（Logs）等数据表，以支持子系统的高效运行。

## 5 软件测试

软件测试是确保系统功能实现正确性和可靠性的关键步骤。本节通过对单元测试、确认测试和非功能测试的详细设计，验证天津大学融合门户系统的各项功能是否符合需求，并评估系统在性能、安全性等非功能性需求方面的表现，循序渐进、逐层递进地开展软件测试。



## 5.1 单元测试

单元测试是对系统中基本模块单元（如过程、函数、类、方法等）进行测试的过程。我们选择用户登录功能作为单元测试的目标，采用白盒测试方法，对代码逻辑的正确性进行详细验证。这里针对社团活动审批服务和绿化类维修申请服务两项服务作为例子具体阐述分析。

### 5.1.1 社团活动审批服务

1. 绘制程序流程图，如下：

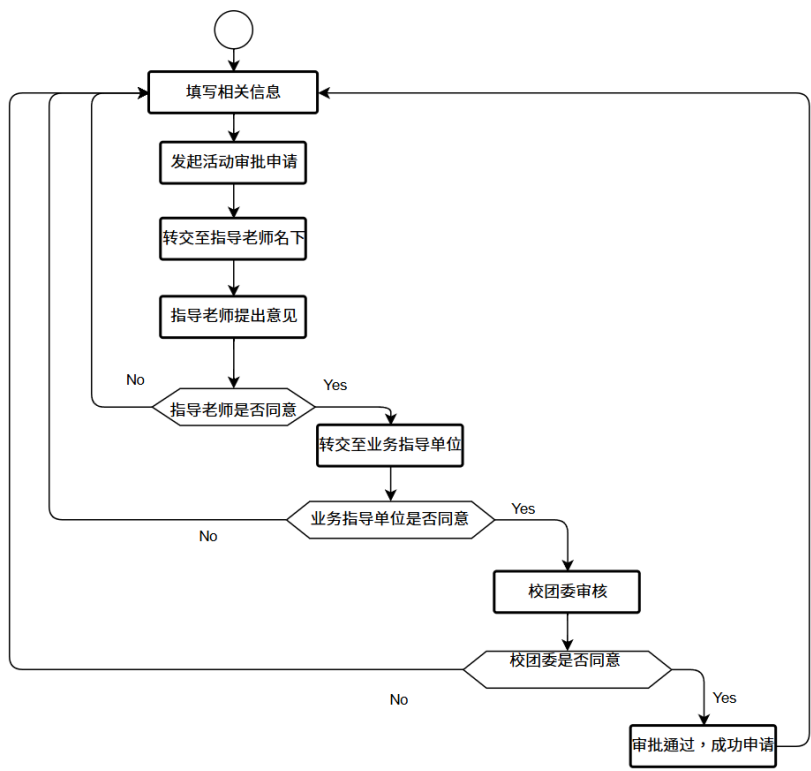


图 5-1 社团活动审批服务程序流程图

2. 依据程序流程图，绘制程序流图：

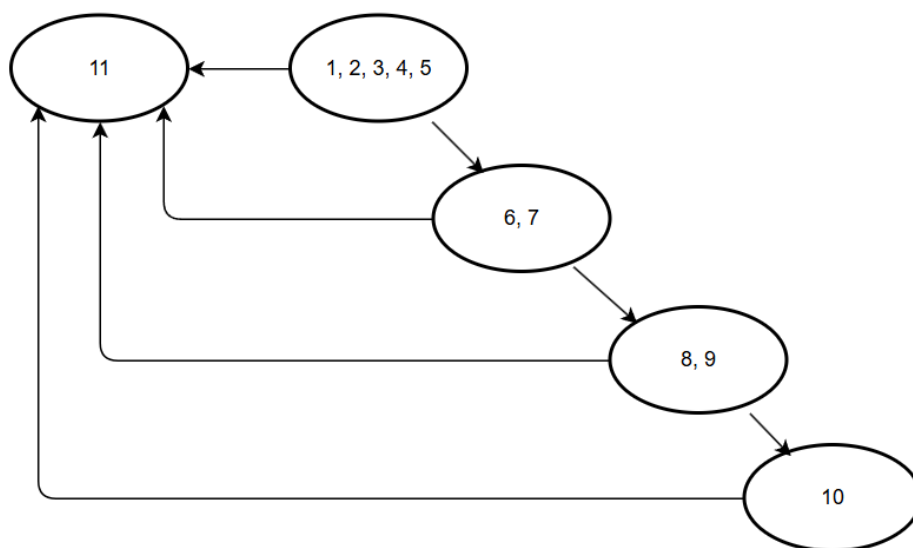


图 5-2 社团活动审批服务程序流程图

### 3. 确定基本路径：

接着确定流图复杂度，计算  $V(G) = E(dges) - N(odes) + 2 = 7 - 5 + 2 = 4$ ，得知共有 4 条基本路径，分别是：

- (1) 1, 2, 3, 4, 5 - 11
- (2) 1, 2, 3, 4, 5 - 6, 7 - 11
- (3) 1, 2, 3, 4, 5 - 6, 7 - 8, 9 - 11
- (4) 1, 2, 3, 4, 5 - 6, 7 - 8, 9 - 10 - 11

### 4. 针对测试路径设计测试用例：

- (1) 1, 2, 3, 4, 5 - 11：指导教师不同意（False），其余取任意值；
- (2) 1, 2, 3, 4, 5 - 6, 7 - 11：指导教师同意（True），业务指导教师不同意（False），其余取任意值；
- (3) 1, 2, 3, 4, 5 - 6, 7 - 8, 9 - 11：指导教师同意（True），业务指导教师同意（True），校团委不同意（False）；
- (4) 1, 2, 3, 4, 5 - 6, 7 - 8, 9 - 10 - 11：指导教师同意（True），业务指导教师同意（True），校团委同意（True）。

此外，实际开发中还需要进行测试用例描述，描述的内容包括用例标识、用例开发者、用例开发日期、测试项、测试输入、前置条件、环境要求、测试步骤、预期输出、用例间依赖性等信息。

### 5. 运行程序检验测试用例

主要分为如下步骤：

- (1) 运行待测试的程序代码；
- (2) 逐个输入测试用例；

- (3) 分析程序的运行路径；
- (4) 如果运行路径与期望路径不一样，则存在缺陷。

## 6. 测试环境

这里列举软件测试中较为常用的测试环境：

- (1) 操作系统：Windows 10
- (2) 开发环境：IntelliJ IDEA 2023、Java 11
- (3) 数据库：MySQL 8.0
- (4) 测试工具：JUnit 5

实际开发过程中，我们还需要进行测试结果分析。如果所有测试用例的实际输出与预期输出一致，说明该模块的功能实现符合设计要求，相关功能通过单元测试。

## 5.1.2 绿化类维修申请服务

### 1. 绘制程序流程图，如下：

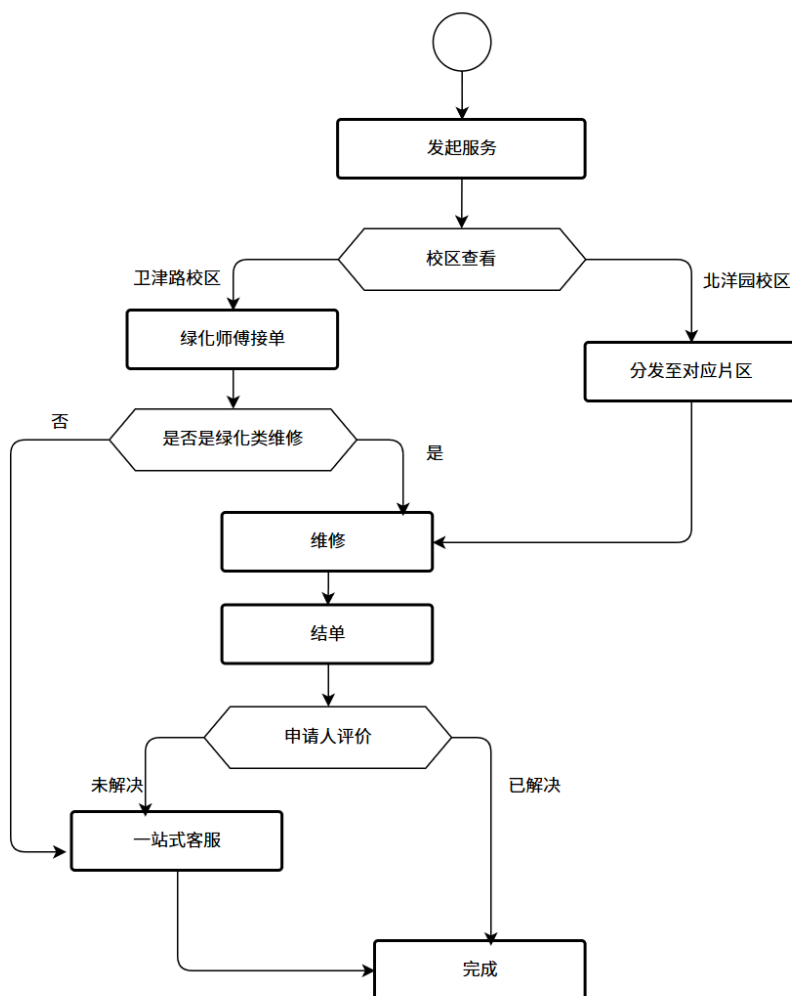


图 5-3 绿化类维修申请服务程序流程图

2. 依据程序流程图，绘制程序流图：

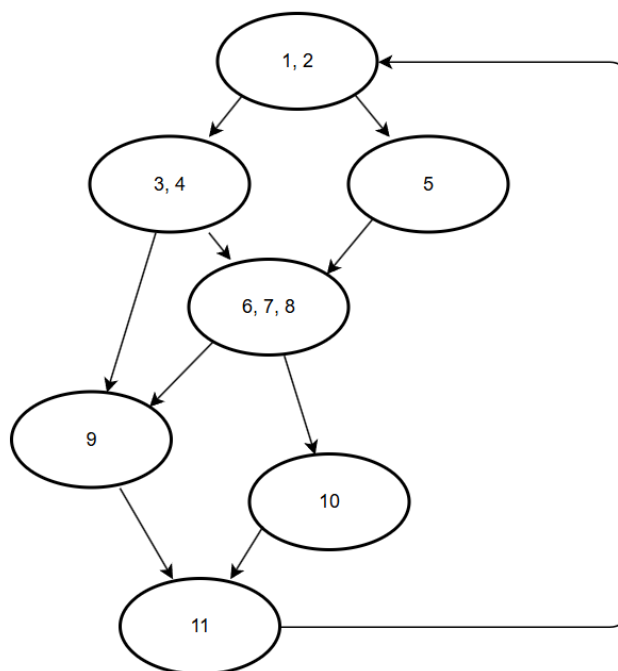


图 5-4 绿化类维修申请服务程序流图

3. 确定基本路径：

接着确定流图复杂度，计算  $V(G) = E(dges) - N(odes) + 2 = 10 - 7 + 2 = 5$ ，得知共有 5 条基本路径，分别是：

- (1) 1, 2, 5 - 6, 7, 8 - 10 - 11
- (2) 1, 2, 5 - 6, 7, 8 - 9 - 11
- (3) 1, 2 - 3, 4 - 6, 7, 8 - 10 - 11
- (4) 1, 2 - 3, 4 - 6, 7, 8 - 9 - 11
- (5) 1, 2 - 3, 4 - 9 - 11

4. 针对测试路径设计测试用例：

- (1) 1, 2, 5 - 6, 7, 8 - 10 - 11：校区为北洋园校区，申请人评价问题已解决，其余取任意值；
- (2) 1, 2, 5 - 6, 7, 8 - 9 - 11：校区为北洋园校区，申请人评价问题未解决，其余取任意值；
- (3) 1, 2 - 3, 4 - 6, 7, 8 - 10 - 11：校区为卫津路校区，维修类型为绿化类维修，申请人评价问题已解决；
- (4) 1, 2 - 3, 4 - 6, 7, 8 - 9 - 11：校区为卫津路校区，维修类型为绿化类维修，申请人评价问题未解决；

- (5) 1, 2 - 3, 4 - 9 - 11: 校区为卫津路校区, 维修类型为非绿化类维修, 其余取任意值。

此外, 实际开发中还需要进行测试用例描述, 描述的内容包括用例标识、用例开发者、用例开发日期、测试项、测试输入、前置条件、环境要求、测试步骤、预期输出、用例间依赖性等信息。最后还应进行测试结果分析。如果所有测试用例的实际输出与预期输出一致, 说明该模块的功能实现符合设计要求, 相关功能通过单元测试。

## 5.2 确认测试

确认测试(黑盒测试)用于验证系统的功能是否满足需求规格说明书中的要求。我们选择课程管理功能进行确认测试, 验证学生和教师用户是否可以正确地管理和查看课程信息。

### 5.2.1 登录功能的测试

#### 1. 测试目标

测试用户登录模块, 验证用户输入用户名和密码后, 系统是否能够正确进行身份验证, 并处理相应的登录结果。

#### 2. 测试用例设计

应用等价类划分法, 将用户名输入划分为 3 类: 学工号存在、学工号不存在, 学工号为空; 将密码输入分为 3 类: 密码正确, 密码错误, 密码为空。因此, 设计 9 个代表性测试用例如下表所示:

表 5-1 登录功能的测试

测试用例 ID	输入数据	预期输出	实际输出	结果
TC001	用户名: “1021244001”, 密码: “12345”	登录成功, 进入主页面	登录成功	Pass
TC002	用户名: “2023244001”, 密码: “wrong”	登录失败, 提示用户名或密码错误	登录失败	Pass
TC003	用户名: “2023244001”, 密码为空	登录失败, 提示密码不能为空	登录失败	Pass
TC004	用户名: “3022244999”, 密码: “12345”	登录失败, 提示用户名不存在	登录失败	Pass
TC005	用户名: “3022244999”, 密码: “wrong”	登录失败, 提示用户名不存在	登录失败	Pass
TC006	用户名: “3022244999”, 密码	登录失败, 提示用户	登录失败	Pass

	为空	名不存在		
<b>TC007</b>	用户名为空，密码：“12345”	提示用户名不能为空	提示用户名为空	Pass
<b>TC008</b>	用户名为空，密码：“wrong”	提示用户名不能为空	提示用户名为空	Pass
<b>TC009</b>	用户名为空，密码为空	提示用户名密码不能为空	提示用户名密码为空，请输入	Pass

### 3. 测试过程

- (1) 初始化测试环境，启动数据库和 Web 服务器。
- (2) 运行各个测试用例，逐一输入不同的用户名和密码组合。
- (3) 记录系统的实际输出，并与预期输出进行对比。

### 4. 测试结果分析

所有测试用例的实际输出与预期输出一致，说明用户登录功能的开发实现符合需求规格说明书的要求。确认测试通过。

## 5.2.2 资讯中心功能的测试

### 1. 测试目标

验证天津大学融合门户系统的资讯中心板块的功能是否正常，检查公告信息发布、编辑、删除等操作是否符合预期。

### 2. 测试用例设计

应用等价类划分法，首先将发布公告时划分为 3 大类共计 10 种情况。其中输入的内容不含特殊字符方面包括：输入完整的标题及正文内容、输入完整的标题但缺少正文内容、输入正文内容但缺少完整的标题、标题及正文内容均缺失。输入的内容含有特殊字符方面包括正文中含特殊字符和标题中含特殊字符和标题正文均含有特殊字符。输入的标题或正文内容超出系统限制（例如，标题超过 50 个字符，正文超过 2000 个字符）方面：这一点共分为 3 种情况讨论（标题过长、正文过程、标题正文均过长）。其次，测试删除动作，这里视为 1 种情况。接着是信息编辑功能，这里视为 1 种情况。最后是权限方面，不同权限的用户（如管理员、普通用户）尝试发布、编辑、删除信息，因为管理员的操作动作已在上文讨论，因此这里只需补充设计普通用户的操作用例，共计 3 种情况。为了简化操作，这里的发布、编辑、删除动作均为合规操作。这里设计 15 个测试用例，如下：

#### (1) 成功发布信息

输入：管理员输入完整的标题、正文内容，并选择适当的类别。

预期结果：信息成功发布并显示在相应的页面上。

(2) 输入的内容不含特殊字符但缺少标题或正文内容（共 2 种情况）

输入：管理员发布信息时，缺少标题或正文内容（只有标题无正文，或只有正文无标题）。

预期结果：系统提示“标题不能为空”或“正文内容不能为空”之类的提示。

(3) 发布空公告

输入：管理员发布信息时，缺少标题及正文内容。

预期结果：系统提示“不能发布空白公告”之类的提示。

(4) 输入内容包含特殊字符（共 3 种情况，其一或均含有特殊字符）

输入：管理员发布信息时，正文内容包含 HTML 标签或特殊字符（如<script>）。

预期结果：系统应过滤掉非法字符，避免 XSS 攻击或显示不正常。

(5) 长标题和长正文（共 3 种情况，其一或均含有超出系统限制）

输入：标题或正文内容超出系统限制（例如，标题超过 50 个字符，正文超过 2000 个字符）。

预期结果：系统应给出合理提示或自动截断超长内容。

(6) 信息编辑功能

输入：管理员选择已发布的消息，编辑其标题或正文内容。

预期结果：成功编辑并保存，页面更新，新的内容显示在页面上。

(7) 信息删除功能

输入：管理员删除某条已发布的消息。

预期结果：信息从系统中删除，并且删除后的页面不再显示该信息。

(8) 信息发布权限控制

输入：没有操作权限的普通用户尝试发布、编辑、删除信息。（3 种情况）

预期结果：普通用户不能发布、编辑或删除他人发布的内容。

### 3. 测试工具及过程

手动测试：模拟不同的内容发布场景。

自动化工具：可以使用 Selenium 等自动化测试工具对发布信息功能进行自动化验证。

### 4. 测试结果分析

所有测试用例的实际输出与预期输出一致，说明用户登录功能的开发实现符合需求规格说明书的要求。确认测试通过。

## 5.3 非功能测试

非功能测试用于验证系统在性能、安全性、可用性等方面的表现。我们选择对系统的并发性能和安全性进行测试，以确保系统能够支持大量用户同时在线操作，并确保数据安全。

### 5.3.1 并发性能测试

#### 1. 测试目标

验证系统在不同负载下的响应速度、稳定性和资源消耗，确保在高并发情况下能够正常处理用户请求，响应时间在可接受范围内并正常运行。

#### 2. 测试工具

**LoadRunner、JMeter:** 模拟大量并发请求，进行负载、压力测试。

**New Relic、AppDynamics:** 监控系统的资源消耗情况。

#### 3. 确定前提条件

软件系统已经完成并且在服务器上安装成功，并准备好测试环境。

#### 4. 测试步骤

- (1) 启动服务器系统；
- (2) 由模拟软件在每个客户机分别依次发送  $n$  个用户登陆请求，并模拟选课过程；
- (3) 监控程序把每个用户的请求和服务器响应过程、时间记录在日志中；
- (4) 对记录结果进行处理，判断是否每个用户均获得成功的服务，并计算平均响应时间。

#### 5. 测试场景

##### (1) 负载测试

- a) 输入：模拟不同数量的用户同时访问系统（如 1000、3000、5000 个用户）。
- b) 预期结果：系统响应时间不超过预设的最大值（如 3 秒内加载页面），系统能够稳定处理并发请求，不出现卡顿、崩溃等现象。

##### (2) 压力测试

- a) 输入：逐步增加请求负载，直到系统达到最大承载能力（如 10000 并发用户），模拟系统过载状态。
- b) 预期结果：系统在高负载下能够处理最大请求量，系统性能下降前应给出明确的提示（如“负载过重，请稍后再试”）。

##### (3) 性能基准测试



- a) 输入：对门户系统的核心功能（如用户登录、服务办理与受理等）进行性能测试。
- b) 预期结果：每个操作的响应时间应符合业务需求，例如登录操作应在 1 秒内完成，服务办理的每个环节将申请流转至正确的对应受理人员。

#### (4) 资源消耗测试

- a) 输入：在高负载的情况下，监控服务器资源（CPU、内存、磁盘 I/O 等）。
- b) 预期结果：系统在高并发条件下不会占用过多的服务器资源，CPU 使用率应小于 80%，内存使用应控制在合理范围内。

### 6. 测试结果分析

最后，测试人员应进行结果分析，分析系统性能、安全性、可用性等表现：比如系统在多少名用户同时在线的情况下能够稳定运行、响应时间控制在多少秒以内、是否达到了性能要求。此外，还应给出后续工作中应针对哪些方面进行提高并提出具体建议等。

## 5.3.2 配置和兼容性测试

### 1. 测试目标

验证系统是否能够在不同的硬件、操作系统、浏览器和设备上正常运行，确保其具备良好的兼容性。

### 2. 测试工具：

- (1) BrowserStack、CrossBrowserTesting：进行浏览器兼容性测试。
- (2) VirtualBox 或 Docker：模拟不同操作系统和硬件配置环境进行兼容性测试。

### 3. 测试场景与过程

#### (1) 浏览器兼容性测试

- a) 输入：在主流浏览器中（如 Chrome、Firefox、Edge、Safari）访问系统。
- b) 预期结果：系统页面在各个浏览器中显示正常，无布局错乱或功能丢失。

#### (2) 操作系统兼容性测试

- a) 输入：在不同操作系统（Windows、Linux、macOS）上运行系统。

- b) 预期结果：系统能在不同操作系统环境下正常运行，用户可以无障碍使用系统的所有功能。
- (3) 设备兼容性测试
  - a) 输入：使用不同的设备（如桌面电脑、平板、智能手机）访问系统。
  - b) 预期结果：系统应自适应不同屏幕大小，保证在手机和平板上的使用体验与桌面端一致。
- (4) 硬件配置兼容性测试
  - a) 输入：测试系统在不同硬件配置下的表现（如 CPU 性能、内存大小、硬盘容量）。
  - b) 预期结果：系统能在不同硬件配置下稳定运行，不会因为硬件性能差异导致运行不流畅。
- 4. 测试结果分析
 

最后，测试人员需对软件系统在兼容和配置性方面的表现进行评估，提供改进策略，以在后续的开发迭代过程中完善软件系统。

### 5.3.3 安全性测试

#### 1. 测试目标

验证系统的安全性，确保数据的保密性、完整性、可用性，确保用户敏感信息在传输过程中不会泄露，防止未经授权的访问和攻击。

#### 2. 测试工具

**OWASP ZAP、Burp Suite：**用于进行安全漏洞扫描和渗透测试。

**SQLMap：**检测 SQL 注入漏洞。

**Wireshark：**监控数据传输，确保加密传输。

#### 3. 测试过程

- (1) 明确潜在的安全隐患，包括容易遭受攻击的内容、可能作为潜在入侵者的人员角色等；
- (2) 明确潜在的入侵行为及时机，例如事务初始化、系统输入、执行存储和检索操作等时候；
- (3) 列出每种潜在安全隐患可能遭到的入侵行为及其可能性；
- (4) 确定风险较高的入侵点，这可通过对包括发生可能性、后果严重程度等因素进行分析得到；
- (5) 根据风险的顺序，设计测试用例来实施安全性测试。

#### 4. 测试用例设计

- (1) 身份验证测试
  - a) 输入：尝试使用不同的身份验证方式（如用户名+密码+验证码、双因素认证）登录系统。
  - b) 预期结果：系统应正确验证身份，未授权用户无法进入系统。
- (2) SQL 注入测试
  - a) 输入：在登录表单、搜索框等地方输入恶意 SQL 语句（如 'OR 1=1 --'）。
  - b) 预期结果：系统应能防止 SQL 注入攻击，不能执行恶意 SQL 查询。
- (3) 跨站脚本（XSS）攻击测试
  - a) 输入：在信息发布或评论区插入 JavaScript 代码。
  - b) 预期结果：系统应能正确处理并过滤恶意脚本，避免 XSS 攻击。
- (4) 权限控制测试
  - a) 输入：使用普通用户身份尝试访问管理员页面或其他权限受限的功能。
  - b) 预期结果：普通用户无法访问管理员权限的功能，系统应阻止非法访问。
- (5) 数据加密与传输安全
  - a) 输入：通过 HTTPS 协议提交敏感数据（如用户登录信息）。
  - b) 预期结果：数据应加密传输，不应出现明文泄漏的情况。
  - c) 测试结果分析

测试人员需要判断系统能否在数据传输的安全性上表现良好、能否有效防御常见的攻击，如中间人攻击和 SQL 注入，确保用户敏感信息的安全。

## 5.4 测试总结

经过单元测试、确认测试和非功能测试的多轮验证，天津大学融合门户系统的主要功能模块已进行了较为系统的测试，开发团队对软件系统在性能、安全性、配置和兼容性等方面的表现也有了较为深入的了解。对于某些方面的不足，后续将不断对其进行修复更新，开发出更完善的软件系统。

## 6 项目总结

本项目通过对天津大学融合门户系统的需求分析、体系结构设计、详细设计以及软件测试，完成了对系统的完整开发流程的学习和实践。本节对整个项目进行总结，评估项目实施过程中的成效、存在的不足之处，并展望未来的改进方向。

### 6.1 系统设计的成效与不足

#### 6.1.1 系统设计的成效

##### 1. 全面掌握软件工程的开发流程

在本次项目中，我们从需求分析入手，逐步进行了体系结构设计、详细设计和软件测试。通过这些环节，我们全面掌握了面向对象的软件设计思想，并能够熟练使用 UML 建模工具进行系统设计。项目中应用的 UML 图，如用例图、顺序图、类图、状态图等，帮助我们更好地理解系统的内部结构和模块间的关系。

##### 2. 功能模块实现

天津大学融合门户系统实现了多个功能模块，包括用户管理、办事大厅、资源共享、通知公告等。各个模块的功能均符合需求规格说明书的要求，且通过了严格的单元测试和确认测试，确保了系统的正确性与稳定性。特别是在权限控制功能中，通过角色的划分（学生、教师、管理员），实现了不同角色对系统资源的访问控制，保证了系统的安全性和可操作性。

##### 3. 系统安全性和性能测试

在系统的非功能性测试中，对并发性能和安全性进行了评估。测试结果表明，系统在 10000 名用户同时在线的情况下能够稳定运行，响应时间符合预期。此外，系统在安全性上表现良好，能够有效抵御常见的网络攻击，如中间人攻击和 SQL 注入，确保了用户数据的安全性。

#### 6.1.2 系统设计的不足

##### 1. 并发性能问题

考虑到天津大学每年本科新生约 5000 人，所以本人推断在性能测试中，系统能够在大约 10000 名用户同时在线时正常运行，但当用户数进一步增加到更多或短时间内进行多次操作（如退选课）时，系统的响应时间明显增加，超出了预期。该问题主要体现在数据库的查询效率和服务器的负载处理能力上。这表明系统在高并发情况下的性能仍需进一步优化。

## 2. 用户界面设计的简化

在本次项目的设计过程中，更多地关注了系统功能的实现，导致用户界面的设计稍显复杂或部分子内容缺失，可以进一步提高美观性和优化用户体验。例如，教务系统中部分课程的授课小结和课程大纲缺失，且内容不居中、比例不够协调，所以看起来不够直观与美观。此外通知公告页面的交互设计较为基础，用户在操作时可能无法获得较好的体验。因此，后续可以在用户界面和用户体验（UI/UX）设计方面做进一步改进。

## 3. 测试覆盖面不足

尽管在软件测试环节设计了详细的单元测试和确认测试，但部分功能的测试覆盖面还不够广泛。例如，系统在移动设备上的兼容性、不同操作系统下的表现等未进行充分的测试。这在实际系统上线后可能导致部分用户在特定设备上遇到问题，影响系统的可用性。

# 6.2 未来改进方向

## 1. 提升系统的并发处理能力

为了解决系统在高并发环境下的性能问题，未来可以考虑引入更多的优化措施。例如通过数据库的优化（如添加索引、优化查询语句等）来提升查询效率；采用负载均衡技术将用户请求分发至多台服务器进行处理，以减轻单台服务器的压力。此外，可以考虑使用 Redis、Memcached 等分布式缓存技术来减轻数据库的负载，加快数据访问速度。

## 2. 优化用户界面和用户体验

用户界面设计是直接影响用户体验的重要因素，未来可以对系统的 UI/UX 进行全面升级。例如，采用 React、Vue 等现代化前端技术来构建动态响应的页面，提升页面的交互性和视觉效果。此外，结合用户反馈，增加一些人性化的功能，如搜索功能、个性化推荐等，以提高用户的使用体验。

## 3. 增强系统的扩展性

未来系统可能需要集成更多功能模块，如可在线查看并下载电子成绩单及在读证明文件（可避免同学们到 1895 行政大楼进行线下办公且无打印

次数限制)等。因此,系统设计需要具备良好的扩展性。在现有的模块基础上,我们可以通过微服务架构的方式来重构系统,使得各个模块可以独立开发、部署和扩展,进一步提升系统的可维护性和可扩展性。

#### 4. 加强测试的覆盖面和自动化测试

为了提升测试的全面性和效率,未来可以引入 Selenium、Jenkins 等自动化测试工具来进行回归测试和兼容性测试,确保系统在各类设备、操作系统下的稳定性。同时,应对系统的边界条件和异常处理进行更多的测试,提升系统的健壮性。

### 6.3 个人总结

通过本次项目的实践,我不仅加深了对软件开发流程和面向对象设计方法的理解,还学会了如何通过测试来验证系统的功能和性能。这次实验报告涵盖了《软件工程》这门课的大部分重要内容,结合我校一年前推出的天津大学融合门户系统,立足专业课程,让我有机会全面复习课上所学知识,在本报告中绘制包括状态图、活动图、顺序图、用例图等相关分析图,全部且超额完成要求的相关图标绘制以及选做任务,并详细撰写实验报告。尽管项目中存在一些不足之处,但这些问题也为开发团队提供了宝贵的改进机会。在未来的系统优化和扩展中,相信开发团队将继续优化系统性能、用户体验以及扩展能力,确保系统能够适应更多的用户需求。我也将结合本人了解和学习的前后端知识、学习融合门户系统的优秀设计,努力学习巩固相关知识,争取在未来能够亲自、独立地完成满足广大用户需求、优雅美观、易用性高的软件系统。