共享花园

系统设计与实现文档

2050633 卢嘉霖

2051475 王浩

2152814 周成杰

2153684 雷扬

2153604 郭晓杰

2152598 郑皓予

2151136 朱开来

2151765 张铭宸

2151268 李昊诚

2151064 余亚辉

**目录**

[1. 共享花园系统需求 1](#_Toc135914124)

[1.1 系统功能性需求 1](#_Toc135914125)

[1.2 系统非功能性需求 2](#_Toc135914126)

[1.3 组织结构 3](#_Toc135914127)

[2. 共享花园系统设计与实现 4](#_Toc135914128)

[2.1 架构设计 5](#_Toc135914129)

[2.2 登录和注册 6](#_Toc135914130)

[2.2.1 登录和注册设计 7](#_Toc135914131)

[2.2.2 登录和注册用例实现 8](#_Toc135914132)

[2.3 论坛子系统 9](#_Toc135914133)

[2.3.1 论坛系统用例设计 10](#_Toc135914134)

[2.3.2 论坛子系统实现 12](#_Toc135914135)

[2.4 维护系统用例 14](#_Toc135914136)

[2.4.1 维护系统用例设计 14](#_Toc135914137)

[2.4.2 维护系统用例实现 16](#_Toc135914138)

[3. 数据库设计 18](#_Toc135914139)

[附录A. 图表索引 30](#_Toc135914140)

# 共享花园系统需求

共享花园是⼀项旨在响应国家节能减排号召的倡议，旨在建设绿⾊低碳校园。它积极促进科学、低碳、⾼效的资源配置模式的发展，以适应⾼等教育的发展形势，并推动⽣态⽂明建设。该项⽬的主要⾏动是将校园内的闲置废弃⼟地改造成共享花园，由所有学⽣参与设计和建造。这个项⽬旨在实现参与式景观设计的⽬标，学⽣可以通过参与获得积分和奖励，如花卉盆栽。针对该项目建⽴的⼀个基于 WEB 的应⽤程序，其中包括建造共享花园的⼯具包和共享花园社区。该数据库管理系统主要针对后台各种数据的增删查改的实现。

## 系统功能性需求

本系统⽀持两种⽤户的访问，⼀种是学校⾥的学⽣，⼀种是校园管理部门的管理⼈员。下面以两种用户类型划分对整个系统支持的功能做出简要的介绍。

学校⾥的学⽣可以在本系统中注册、登陆账号，申请创建、修改、维护或铲除 花园，浏览各个花园相关信息并打分与评论，在论坛中互动，应聘志愿者，同 时也能举报不良内容等。

校园管理部门可以处理学⽣创建、修改、维护或铲除花园的申请，同时也能发 布志愿者招募信息，给予花园贡献者活动积分，审核并删除被举报不良内容等。

将上述提到的功能进行简要的整合后，下面以不同的功能点划分列出该系统支持的一些业务：

* ⽤户账户管理：应当允许⽤户创建和管理他们的账户，包括⽤户个⼈资料、偏好设置和安全设置。不同类别的账号应当能够灵活管理。
* 花园设计和规划：应当提供⼯具让⽤户设计和规划⾃⼰的花园空间， 包括植物选择、布局和灌溉等功能。花园的相关信息包括花园大小、花园位置等应当在数据库中清晰记录并能够随时查看。当需要修改时，用户也应当能够迅速地对自己花园进行管理。
* 志愿者系统：对于需要志愿者参与管理维护的花园，提供志愿者招募的功能。数据库中应当分别记录志愿活动和对应的一些志愿招募和志愿申请，每一次志愿活动的进行受到花园主人的监管。
* 社区参与：应当拥有⿎励社区参与和协作的功能，⽐如活动、论坛和社交分享等。当用户希望参观其他用户的花园时，应当提供相应的申请功能；当用户进行评论等社区行为时，应当受到管理员的监管，例如添加相应的举报机制。
* 反馈和评级系统：实现该系统系统应当能让⽤户对平台上的花园和资源 进⾏反馈和评级。
* 游戏化和奖励机制：应当添加游戏化元素和适当奖励，以激励⽤户参与和互动，例如积分、成就、 徽章和排⾏榜。

## 系统非功能性需求

本节列出了描述系统质量和性能的⾮功能性需求。⾮功能性需求是根据项⽬范围和利益 相关者的期望推导出来的。它们按照系统的主要属性进⾏分类。

>可⽤性

* 系统应提供⼀个⽤户友好和直观的界⾯，易于导航和操作。
* 系统应提供清晰和⼀致的指⽰、标签和反馈给⽤户。
* 系统应提供帮助和⽂档功能，解释系统的功能和使⽤⽅法。
* 系统应⽀持多种语⾔，包括英语和中⽂。

>可靠性

* 系统应确保存储和显⽰的数据的准确性和完整性。
* 系统应防⽌未经授权的访问和修改数据。
* 系统应优雅地处理错误和异常，并向⽤户提供适当的错误信息。
* 系统应能从故障中恢复，并尽快恢复正常运⾏。

>效率

* 系统应在合理的时间范围内响应⽤户请求。
* 系统应优化⽹络带宽和服务器资源的使⽤。
* 系统应⽀持多个⽤户同时访问，⽽不影响系统的性能或功能。

>可维护性

* 系统应是模块化和有良好⽂档的，以便于未来的变更和增强。
* 系统应遵循编码标准和最佳实践，以确保代码的可读性和⼀致性。
* 系统应使⽤开源技术和框架，以减少对专有软件的依赖。

## 组织结构

列出文档的组织结构。

第一章：总体设计和需求分析

第二章：系统用例与活动图

第三章：ER图和关系模式的设计

………….

附录A：是本文档的图表索引。

# 共享花园系统设计与实现

该项目主要通过将校园内的废弃土地改造成共享花园，由所有学生参与设计和建造来实现参与式景观设计的目标。学生可以通过参与获得积分和奖励，如花卉盆栽。为了方便学生和管理员的交互，该项目基于Web进行开发，主要包括登录和注册系统、维护系统、论坛系统、访问系统和创建与修改系统等五个主要用例。

登录和注册系统是该项目的基础，它允许用户注册账户并登录进入系统。维护系统主要由管理员负责，包括查看和管理花园的状态，以及安排维护和修剪花园等。论坛系统提供给用户一个交流平台，可以在其中分享和交流关于共享花园的信息和经验，对于不符合网络文明的发言或评论，用户可对其发起举报，由管理员作出处理。访问系统允许用户浏览共享花园的信息，包括花园的位置、开放时间和预约情况等。创建和修改系统是学生参与共享花园建设的核心，它允许学生选择花园、设计方案和提交预约申请等，同时也允许管理员审核和处理预约申请。

除了上述的主要用例，该项目还包括了一些辅助用例，例如志愿者系统、积分系统和奖励系统等。志愿者系统可以帮助用户增加志愿管理并提升整个校园的共享花园参与感，包括志愿发布、志愿申请、志愿实行和志愿奖励等。积分系统和奖励系统激励学生参与共享花园的建设，为他们提供额外的动力和奖励。

附言：1.本文档中给出的用例截图均由StarUML绘制

2.本文档中的用例图是反应系统的大致逻辑，并且是初步设计，并不要求后续开发一定的遵循所有细节

下面首先给出总的用例设计图：

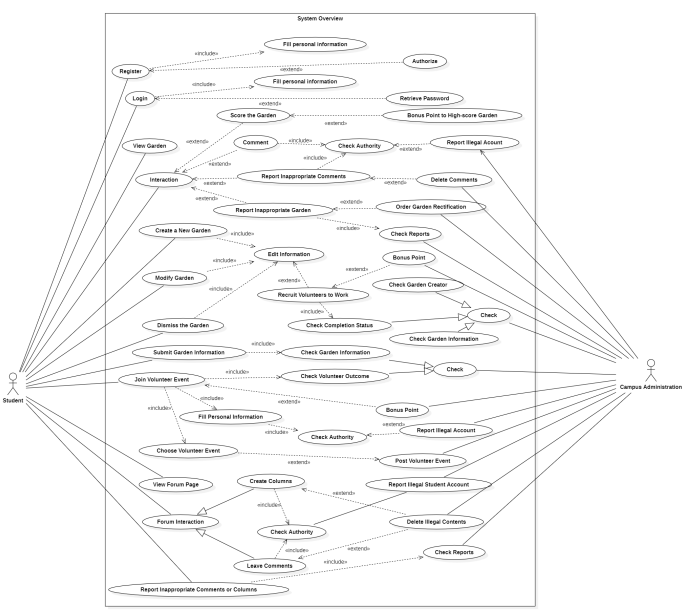


图 总用例图 1

## 架构设计

根据整个系统的用例图对系统进行架构设计，对于该共享花园数据库管理系统，选择分层架构可以提高系统的可扩展性、安全性、可测试性、可维护性等方面的优势。例如，可以将数据库层、业务逻辑层和用户界面层分离，以便更好地管理和维护不同层次的代码和功能。这样可以更好地控制访问和保护敏感信息，提高系统的安全性。同时，每个层次都可以独立测试和验证，从而降低了测试的复杂度和成本，提高系统的可测试性。此外，分层架构还可以提供更好的文档和注释，以帮助开发人员更快地理解和修改代码，提高系统的可维护性。综上所述，选择分层架构可以为共享花园数据库管理系统提供更好的可扩展性、安全性、可测试性和可维护性等优势。

下面给出具体的架构：

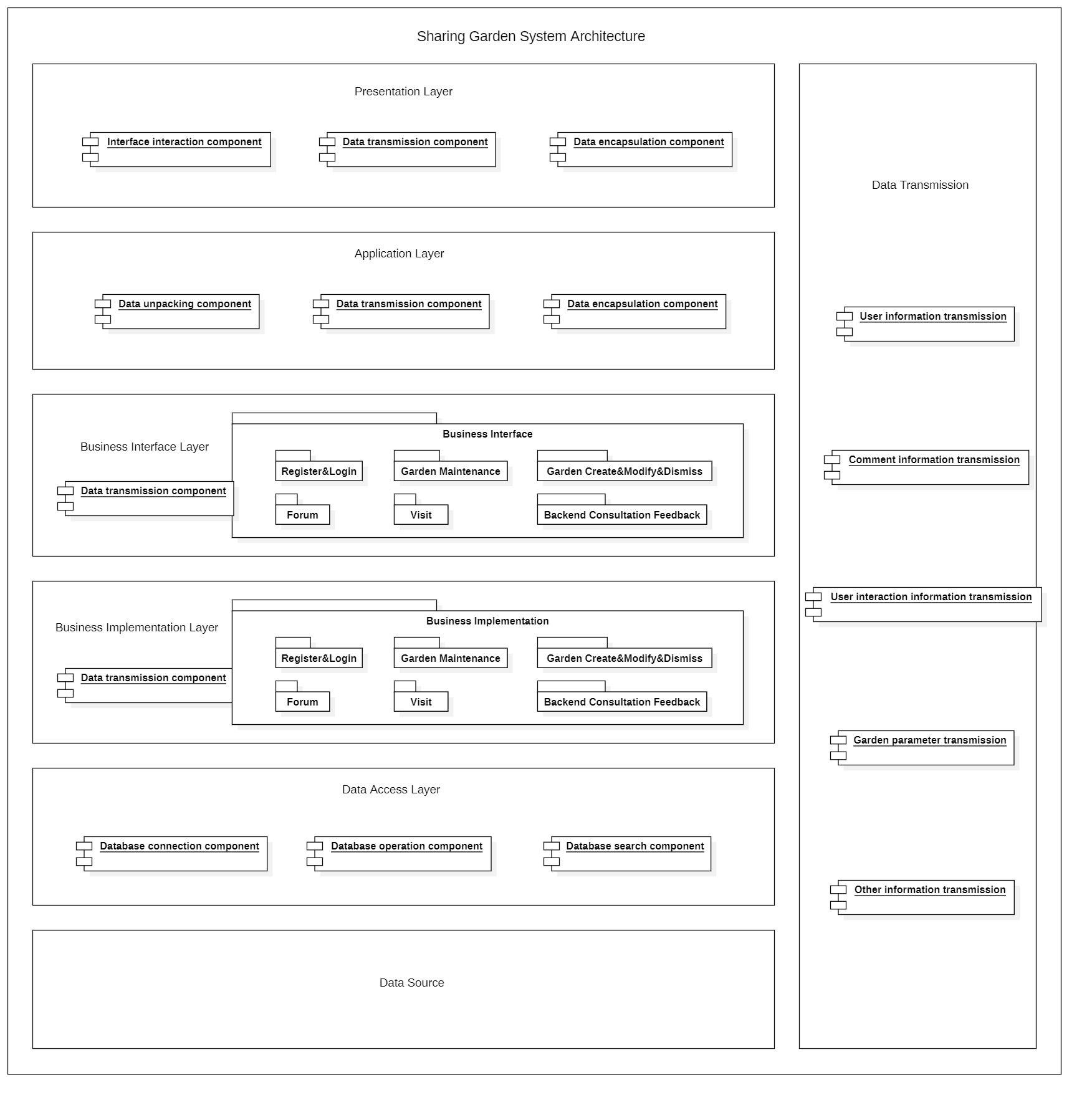


图 2-1 系统架构设计 1

该架构采用传统的分层设计，整体包括表示层、业务逻辑层、数据访问层，在细节上又进行细化，比如将业务逻辑层再分为业务界面和业务实现等。各个层次具体内容如上。

## 登录和注册

登录和注册是共享花园项目中非常重要的用例，它们是用户进入系统的必要步骤。用户需要先注册账户，然后使用用户名和密码登录系统。在注册和登录过程中，系统需要验证用户输入的信息并确保其安全性。

在注册用例中，用户需要输入用户名、密码和其他必要信息，然后点击注册按钮，系统会验证这些信息并确保其有效。如果输入的信息无效或用户名已经被注册，用户将会被提示重新输入或重新注册。一旦用户成功注册，系统会自动将其跳转至登录页面。

在登录用例中，用户需要输入正确的用户名和密码，系统会验证用户信息并允许用户进入系统。如果输入的用户名或密码不正确，用户将会被提示重新输入。如果用户多次输入错误的信息，系统会锁定用户账户以保护其安全性。

### 登录和注册设计

该用例为系统的一般通用用例，大致设计逻辑并无特殊，下面给出简单的登录和注册用例的相关动作序列

表格 2-2 登录和注册功能的动作序列 1

|  |  |
| --- | --- |
| 动作序列 | 描述 |
| 录入信息 | 前置条件：用户已进入注册页面或登录页面 |
| 系统验证 | 验证录入信息是否符合已定的规则 |
| 点击注册 | 等待系统响应🡪成功或失败，给出相应的信息  这里给出各种失败的替代流程：  1. 输入的信息无效，用户被提示修改信息  2. 用户名已经注册，用户被提示重新输入信息  3. 用户名或密码不正确，用户被提示重新输入信息  4. 用户多次输入错误信息，系统会锁定账户 |
| 登录输入 | 前置条件：用户已经成功注册 |
| 系统验证 |  |
| 进入主页 | 前置条件：用户已经登录成功 |

根据上面的动作序列，简单地将登录和注册子系统的业务逻辑进行整理，检查用例规约的正确性和完善性，分析用户使用该子系统的整个流程，对系统做出鲁棒性分析：

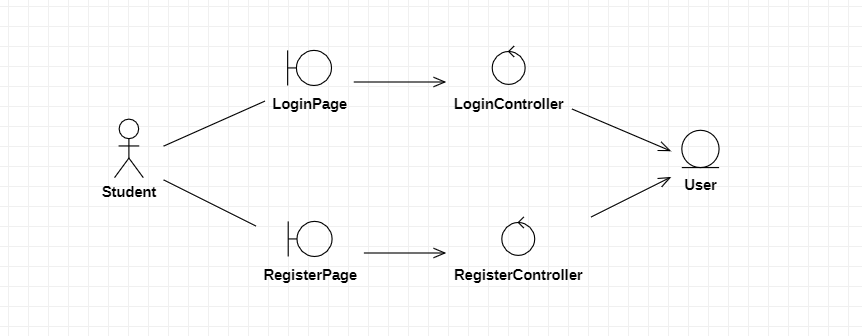


图 2-2-1 登录和注册子系统鲁棒性 1

### 登录和注册用例实现

根据上面的动作序列表和具体的用例要求，在该子系统中将相关数据进行归类整理，对每个类中需要的数据做出定义，下面给出具体的类图：

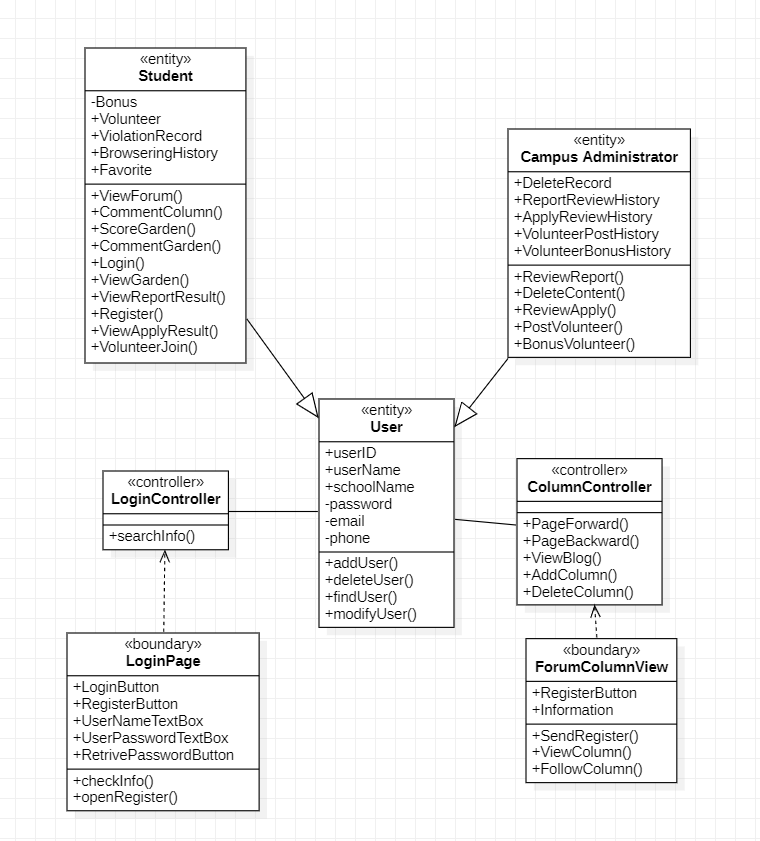


图 2-2-2 登录和注册功能实现 1

该类图给出登录和注册子系统涉及到的关键数据类，其中包括视图类LoginPage等和数据类student等，在各个类图的关系联系中还涉及到应用接口LoginController实现页面上交互和数据库中数据操作的链接。

## 论坛子系统

该子系统为用户提供了一个交流平台，可以在其中分享和交流关于共享花园的信息和经验。该论坛系统主要管理所有的评论、发言等内容，包括举报功能、文章分享、用户间的交流等功能。

在论坛系统中，用户可以发布文章和评论，与其他用户进行交流。用户可以在论坛中分享自己的经验和知识，也可以向其他用户提出问题和寻求帮助。用户可以浏览和回复其他用户发布的文章和评论，并进行点赞和举报等操作。

在举报功能中，用户可以对不当言论或违反论坛规则的内容进行举报，管理员可以对举报内容进行审核，并对不当言论进行处理。管理员还可以管理和维护论坛系统，包括对用户的管理和权限控制等。在文章分享中，用户可以将自己的文章发布到论坛中，与其他用户分享自己的经验和知识。其他用户可以浏览和回复这些文章，并进行点赞和分享等操作，从而促进共享花园的建设和发展。

在用户间的交流中，用户可以通过私信或评论等方式与其他用户进行交流。这些交流可以增进用户之间的了解和合作，从而推动共享花园的建设和发展。

### 论坛系统用例设计

该部分主要包括两个方向的功能，一是用户间的交互、二是对系统内不当行为的处理，这里具体为博客和评论浏览互动、举报功能。

在交流互动中应当包括浏览博客文章、评论、举报等用例，再对举报用例进行详细的流程分析。下面给出两个部分的动作序列：

表格 2-3-1 交互功能的动作序列 1

|  |  |
| --- | --- |
| 动作序列 | 描述 |
| 学生登录进入主页 | 涉及到登录与注册系统 |
| 浏览博客或者评论 | 具体的动作可以包括：  1. 学生可以通过滚动界面浏览不同帖子的关键词  2. 学生可以选择感兴趣的帖子并点击进入  3. 学生可以选择屏蔽不感兴趣的帖子 |
| 选择交互动作 | 包括：评论、举报等 |
| 系统验证 | 这里主要验证学生账户的有效性，如果通过，学生可以编辑留言或帖子内容；  在验证学生的操作内容是否合法 |
| 学生完成操作 |  |
|  | 这里给出其他相关细节：  替代流程：  1. 学生进入论坛后发现没有感兴趣的帖子并退出论坛  2. 学生在浏览过程中选择切换账户  异常流程：  1. 学生反复选择登录账户，系统会提示失败  2. 学生浏览完内容后选择下拉界面，系统会提示没有更多内容  3. 学生在短时间内频繁发布评论或帖子，系统会认为是垃圾邮件并限制发布评论或帖子的时间  4. 学生发布的评论或帖子被举报多次且存在违规行为，学生账户将被锁定，无法正常发布评论或帖子 |

表格 2-3-2 举报功能的动作序列 1

|  |  |
| --- | --- |
| 动作序列 | 描述 |
| 交互中选择“举报” | 涉及到上面给出的交互过程 |
| 填写举报原因 | 给出的举报需要受到系统或管理员审核，由管理层做出处理操作 |
| 系统反馈 | 1. 如果帖子或评论违反论坛规则，将被删除，并且学生将会收到通知  2. 如果帖子或评论没有违反论坛规则，将会被转交给校园管理者 |
| 后续处理 | 1.如果举报有效，发布违规评论或帖子的用户将会被记录  2.如果发布违规评论或帖子的用户多次违规，其账户将被封锁一段时间，期间无法发布评论或帖子 |
|  | 这里给出其他相关细节：  替代流程：  1. 如果学生在提交举报前想要撤回举报，可以这样做  2. 学生可以查看其举报的进展情况  异常流程：  1. 如果学生点击“举报”按钮但未填写举报原因，举报将不会被提交  2. 如果学生在短时间内提交多个无效的举报，其举报权限将被限制一段时间  3. 如果在发布、评论和举报权限被限制的期间内再次被举报，校园管理者将无法再次限制其权限。 |

针对上述动作序列，对该子系统进行鲁棒性检查：

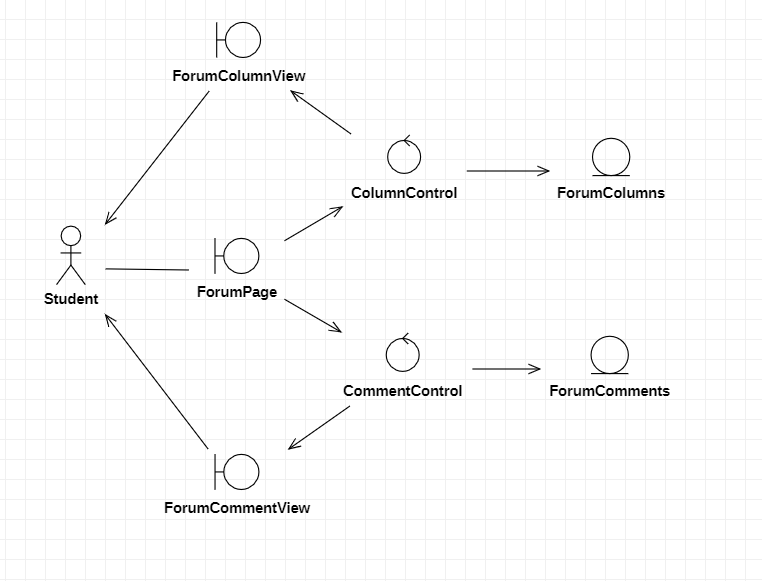


图 2-3-1 鲁棒性—论坛子系统1

并针对论坛系统中的一个重要功能“举报”进行单独的鲁棒性分析：

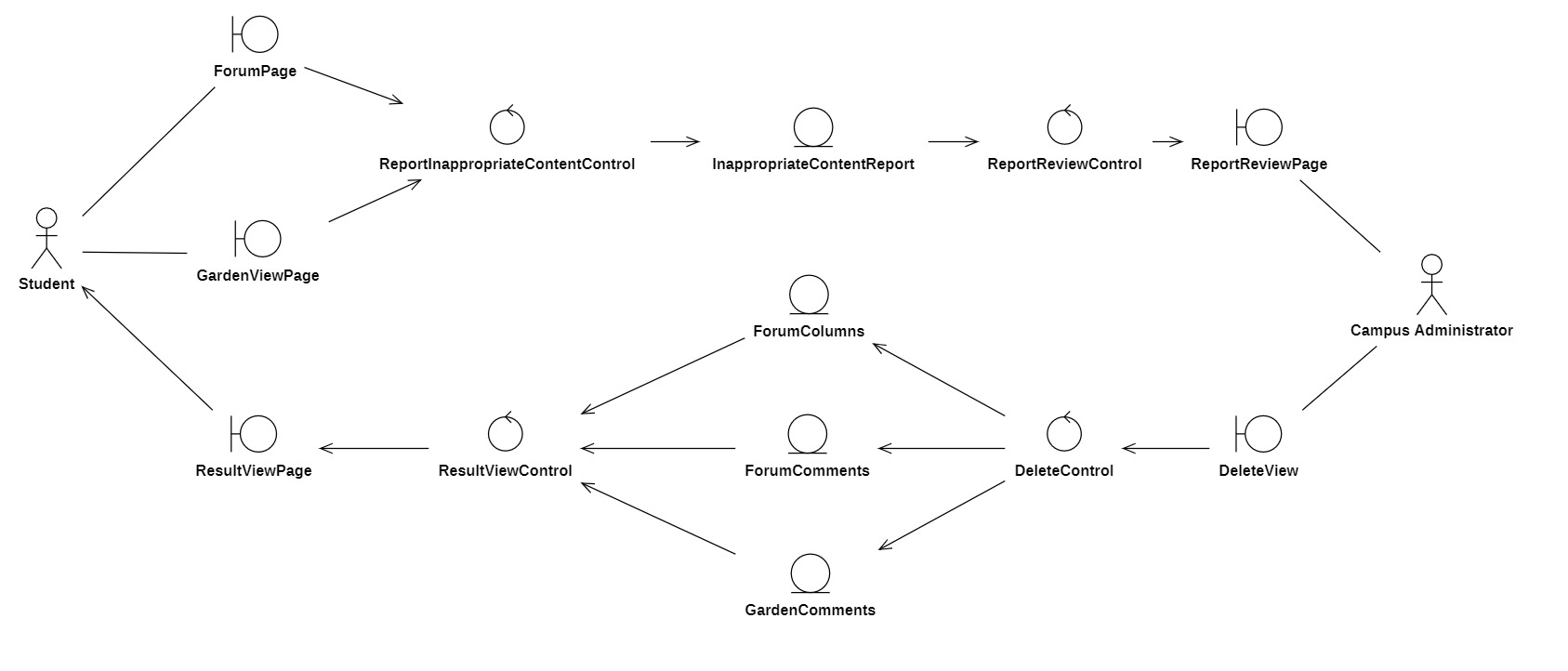


图 2-3-2 鲁棒性—举报功能 1

### 论坛子系统实现

一个论坛子系统是一个在线社交平台，允许用户发表帖子、回复帖子、查看帖子，并与其他用户进行互动和交流。一个典型的论坛子系统通常包括多个模块和功能，这里的子系统其实是相对而言抽象的的，主要管理评论、文章等。

设计该子系统同样应该考虑到数据库数据和视图控制逻辑等，将这样的要求结合架构中划分的层次将需要的功能模块分到不同的层次中，如视图类中主要分为：ForumColumnView、ForumPage、ForumCommentView，类似地继续完成交互控制和底层数据的设计。具体的类图设计如下：

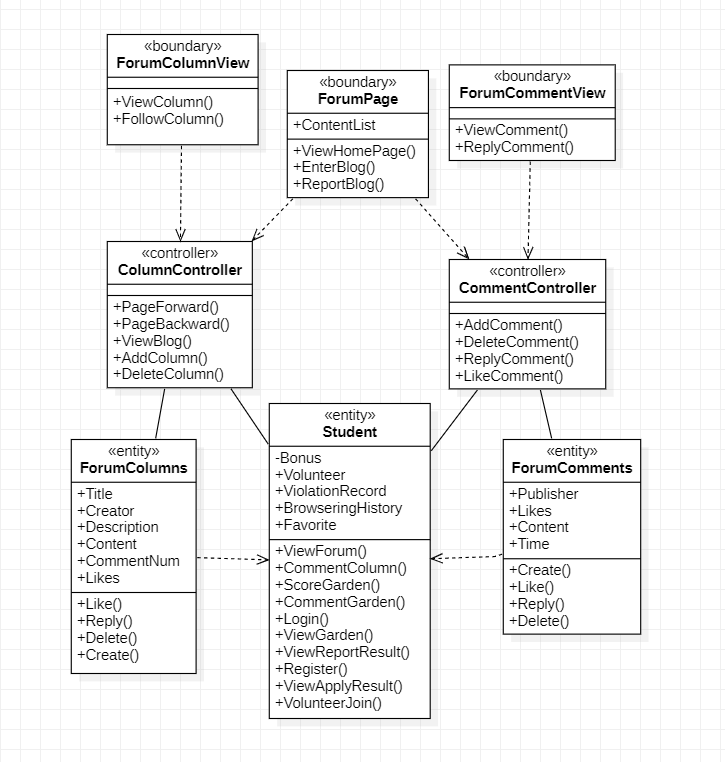


图 2-3-3 论坛功能实现 1

为方便处理整个论坛系统中“举报”功能，理解在整个举报过程中各个参与者的数据和交互处理逻辑，单独对“举报”功能进行类图分析，展示在举报过程中涉及到的各个子系统和不同的架构层次：

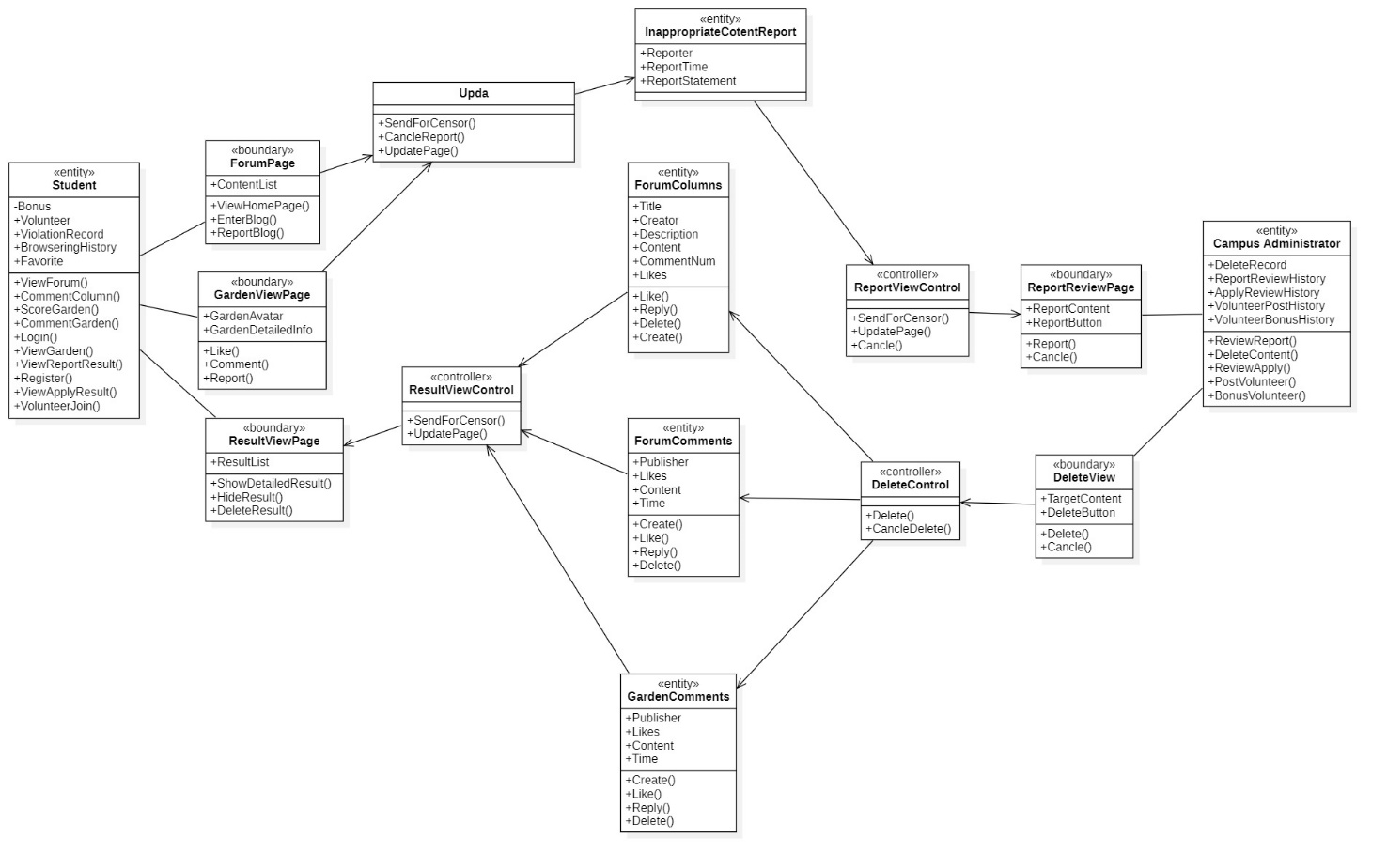


图 2-3-4 类图—评论 1

## 维护系统用例

这个维护系统旨在通过校园内的学生志愿者来维护和保养花园。学生可以提交与花园维护相关的信息，如老化、缺陷或需要额外花盆等，并在校园管理人员审核后发布志愿者招募信息。学生可以在系统接口上注册成为志愿者并选择感兴趣的志愿者活动，同时系统会自动检查学生的个人信息和权限，以确保他们适合参加志愿者活动。该系统允许学生自愿参与花园维护和保养工作，并为他们提供相应的活动积分作为奖励。校园管理人员会检查参与花园维护的学生志愿者的服务输出，并根据他们的表现奖励活动积分。这个积分系统可以激励学生积极参与志愿者活动，并提高他们的参与度和贡献度。

### 维护系统用例设计

该维护系统主要的业务核心围绕志愿者，整体的动作包括发现需求、提交志愿招募、用户申请等多个子用例，下面是设计的相关动作序列：

表格 2-4 维护功能的动作序列 1

|  |  |
| --- | --- |
| 动作序列 | 描述 |
| 发现需求 | 学生填写与花园当前老化、缺陷或需要额外花盆等相关的信息并提交 |
| 审核 | 校园管理人员结合实地信息审核学生提交的花园相关信息 |
| 发布招募 | 由花园所有者发布管理员审核通过的志愿招募信息 |
| 选择志愿 | 其余用户在系统上选择感兴趣的志愿活动进行申请 |
| 审核申请信息 | 由系统检验申请人提供的信息是否正确且符合规定，由花园拥有者决定是否接受申请；若招募满足条件后，系统或花园拥有者删除招募信息 |
| 奖励 | 检查参与花园维护的学生志愿者的服务输出并根据其表现奖励活动积分 |
|  | 这里给出其他的相关细节：  - 学生在填写相关信息后没有立即提交，系统会自动保存草稿；  - 成为志愿者活动的成员后，学生申请退出活动，并经过系统批准；  - 如果花园维护工作量很小且申请人愿意自行完成工作，则校园管理人员无需招募额外的志愿者；  - 如果申请人在进行花园维护和保养活动时中途退出，校园管理人员可以放弃志愿者招募信息；  - 如果校园管理人员发现学生提交的花园相关信息不实，则不发布相关志愿者招募信息；  - 学生参加志愿者活动但拒绝工作，不会对花园维护产生影响，校园管理人员不会奖励积分并记录一次不当行为；  - 如果发布了志愿者信息，但没有足够的人愿意担任志愿者，则花园维护和保养计划可能失败；  - 如果志愿者处理工作不当，则花园维护和保养计划也可能失败。 |

上面的动作序列展示了用户在系统中使用志愿功能时可能会遇到的场景，结合用例图的设计，对该子系统进行鲁棒性分析，包括查看花园和修改花园：

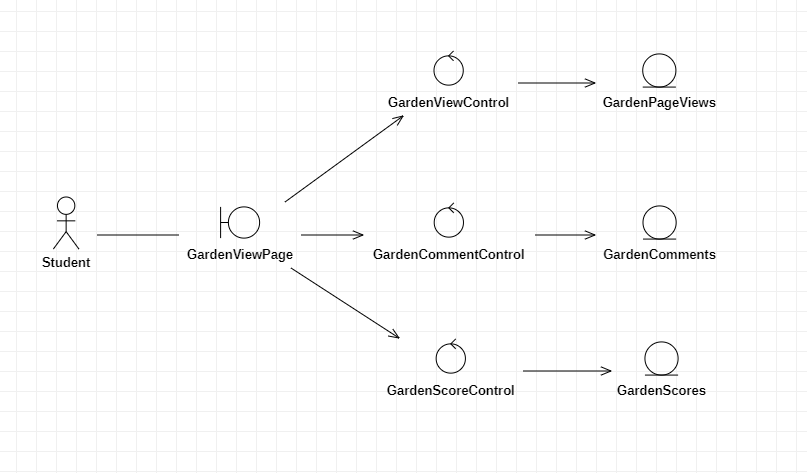


图 2-4-1鲁棒性—查看花园 1

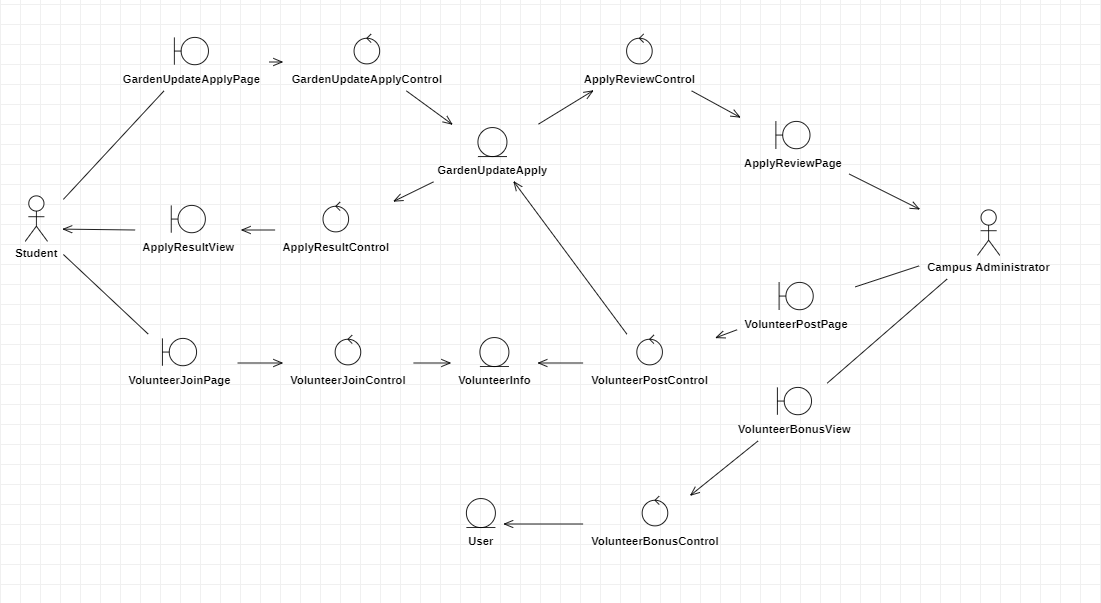


图 2-4-2 鲁棒性—修改花园 1

### 维护系统用例实现

以志愿系统为主要内容绘制相关的类图，主要展示志愿活动涉及到的底层数据和交互逻辑，相对于前面的系统，该子系统相对更为复杂，下面针对该系统主要的两个功能给出相应的类图：

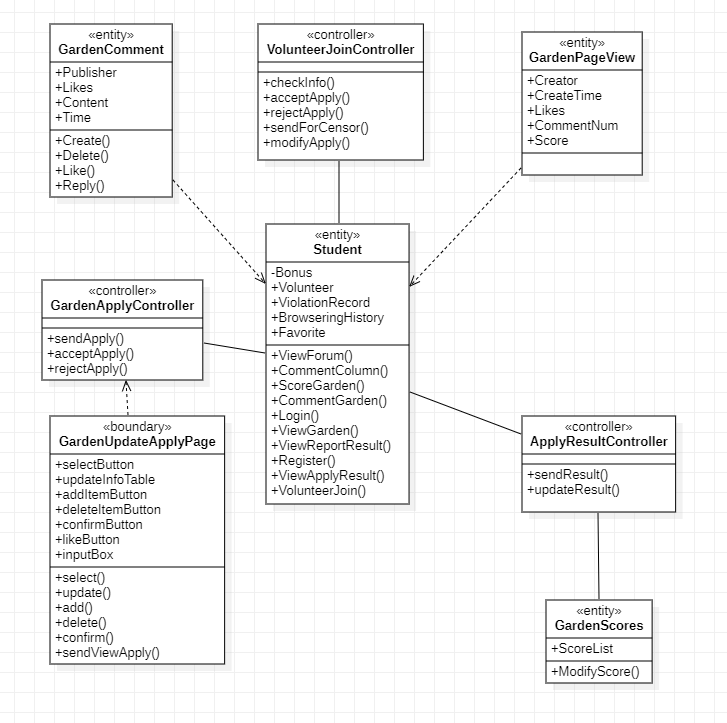


图 2-4-3 类图—查看花园1

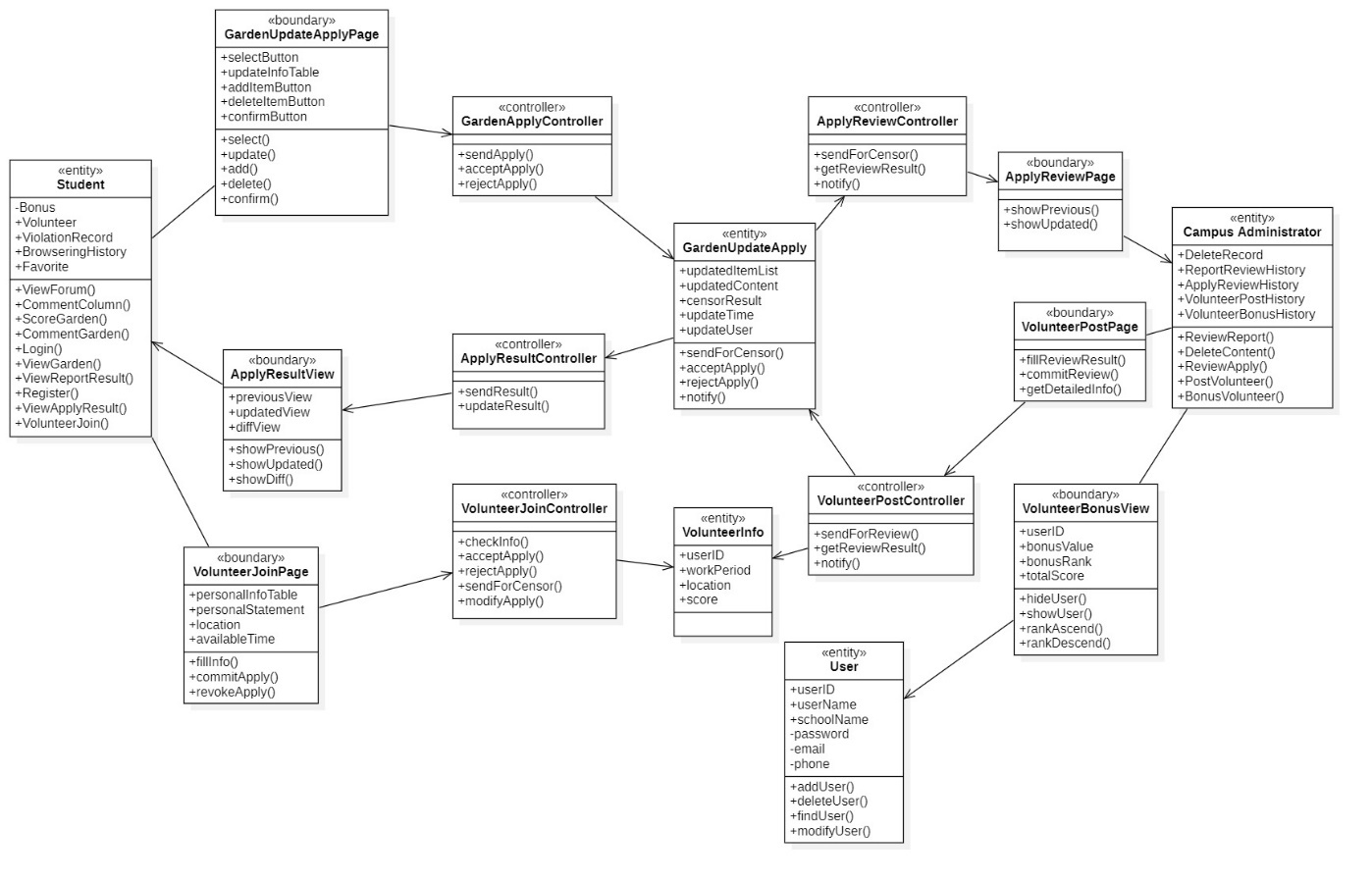


图 2-4-4 类图—修改花园 1

# 数据库设计

根据上面给出的用例设计，对需要的实体和关系做出下面的初步分析：

1. 实体

- 用户：包括学生和志愿者，记录用户的基本信息，如用户名、密码、姓名、联系方式等。

- 花园：记录花园的基本信息，如花园名称、位置、面积、种植植物种类等。

- 活动：记录花园创建、修改和养护活动的基本信息，如活动名称、时间、地点、描述等。

- 评论：记录用户的评论信息，包括评论内容、评论时间、评论者、被评论的花园或活动等。

2. 关系

- 用户与花园：一个用户可以创建或参与多个花园的创建、修改或养护活动，一个花园也可以有多个创建者或参与者。

- 花园与活动：一个花园可以有多个创建、修改或养护活动，一个活动也只能属于一个花园。

- 用户与活动：一个用户可以参与多个创建、修改或养护活动，一个活动也可以有多个参与者。

- 花园与评论：一个花园可以有多条评论，一条评论只能属于一个花园。

- 活动与评论：一个活动可以有多条评论，一条评论只能属于一个活动。

根据上述实体和关系，可以初步的设计相关的类，下面给出设计的ER图：

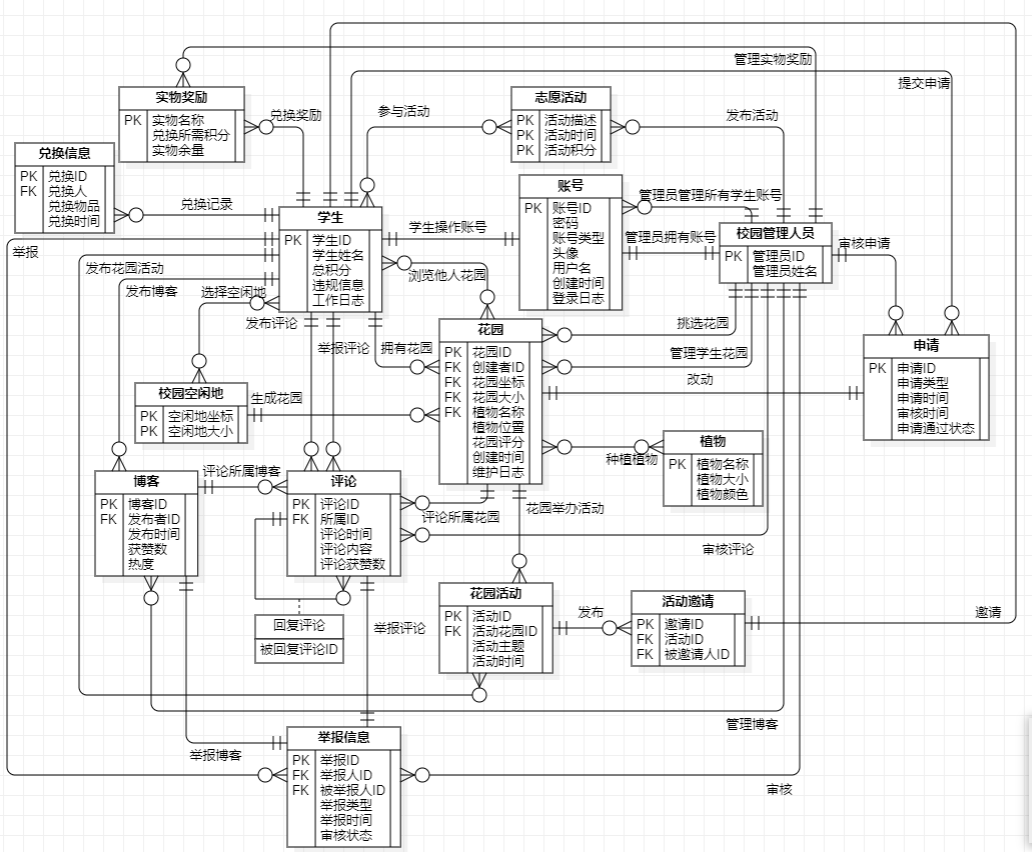
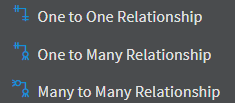


图 3 ER图 1

在该ER图中，实体之间的关系遵循UML一般规则：



该共享花园系统的数据库设计涉及到多个实体和关系，通过ER图设计可以表示出它们之间的关系和依赖。若将学生和花园实体间的从属关系作为例子，我们可以做出如下的规定：花园是主实体，用户是从实体，因为一个花园可以有多个创建者或参与者，但一个用户只能创建或参与一个花园。

根据上面的ER图并考虑各个实体之间的关系和约束条件，我们给出如下的关系表和关系模式：

**表 1 账户(Account)**

总体关系表 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | account\_id | char(10) | 账号ID | PK |
| 2 | password | varchar(20) | 密码 |  |
| 3 | stu\_id | char(10) | 学生ID | FK, Student(stu\_id) |
| 4 | account\_type | varchar(20) | 账号类型 |  |
| 5 | create\_time | datetime | 创建时间 |  |
| 6 | login\_info\_id | char(10) | 登录ID | FK, Login\_Info(login\_info\_id) |

**表 2 账户概要(Account\_Profile)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | account\_id | char(10) | 账号ID | PK |
| 2 | avatar | varchar(255) | 头像 |  |
| 3 | otherinfo | text | 其他信息 |  |
| 4 | registration\_time | datetime | 注册时间 |  |

**表 3 学生(Student)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | stu\_id | char(10) | 学生ID | PK |
| 2 | name | varchar(20) | 学生姓名 |  |
| 3 | account\_id | char(10) | 账号ID | FK, Account(account\_id) |
| 4 | point | int | 总积分 |  |

**表 4 管理员(Administer)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | admin\_id | char(10) | 管理员ID | PK |
| 2 | admin\_name | varchar(20) | 管理员姓名 |  |
| 3 | account\_id | char(10) | 账号ID | FK, Account(account\_id) |

**表 5 登录(Login\_Info)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | login\_info\_id | char(10) | 登录ID | PK |
| 2 | account\_id | char(10) | 账户ID | FK, Account(account\_id) |
| 3 | login\_time | datetime | 登录时间 |  |
| 4 | login\_location | varchar(15) | 登录位置 |  |

**表 6 违规信息(Violation\_Info)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | account\_id | char(10) | 账户ID | PK |
| 2 | violation\_description | text | 违规描述 |  |
| 3 | violation\_time | datetime | 违规时间 |  |

**表 7 博客(Blog)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | blog\_id | char(10) | 博客ID | PK |
| 2 | account\_id | char(10) | 账号ID | FK, Account(account\_id) |
| 3 | blog\_type | varchar(20) | 博客类型 |  |
| 4 | title | varchar(50) | 博客标题 |  |
| 5 | release\_time | datetime | 发布时间 |  |
| 6 | interactive\_rating | float | 互动评分 |  |
| 7 | is\_legal | boolean | 合法性 |  |

**表 8 博客评分(Blog\_Ratings)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | blog\_rating\_id | char(10) | 博客评分ID | PK |
| 2 | blog\_id | char(10) | 博客ID | FK, Blog(blog\_id) |
| 3 | account\_id | char(10) | 账号ID | FK, Account(account\_id) |
| 4 | rating\_value | float | 评分值 |  |
| 5 | rating\_time | datetime | 发布时间 |  |

**表 9 博客评论(Blog\_Comment)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | comment\_id | char(10) | 评论ID | PK |
| 2 | blog \_id | char(10) | 博客ID | PK FK, Blog(blog\_id) |
| 3 | comment\_time | datetime | 评论时间 |  |
| 4 | comment\_content | varchar(200) | 评论内容 |  |
| 5 | commentator\_id | char(10) | 回复ID | FK, Account(account\_id) |

**表 10 申请(Apply)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | apply\_id | char(10) | 申请ID | PK |
| 2 | apply\_type | varchar(20) | 申请类型 |  |
| 3 | apply\_time | datetime | 申请时间 |  |
| 4 | account\_id | char(10) | 账号ID | FK, Account(account\_id) |
| 5 | is\_approved | boolean | 申请通过状态 |  |

**表 11 举报(Report)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | report\_id | char(10) | 举报ID | PK |
| 2 | reporter\_id | char(10) | 举报者ID | FK, Account(account\_id) |
| 3 | reported\_id | char(10) | 被举报者ID | FK, Account(account\_id) |
| 4 | reported\_content\_id | char(10) | 被举报内容ID |  |
| 5 | report\_type | varchar(20) | 举报类型 |  |
| 6 | report\_time | datetime | 举报时间 |  |

**表 12 举报审核记录(Report\_approval\_history)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | approval\_id | char(10) | 审核记录ID | PK |
| 2 | report\_id | char(10) | 举报ID | FK, Report(report\_id) |
| 3 | approval\_result | varchar(20) | 审核结果 |  |
| 4 | approval\_time | datetime | 审核时间 |  |

**表 13 花园(Garden)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | garden\_id | char(10) | 花园ID | PK |
| 2 | owner\_id | char(10) | 所有者ID | FK, Account(account\_id) |
| 3 | create\_time | datetime | 创建时间 |  |
| 4 | garden\_size | int | 花园大小 |  |
| 5 | garden\_pos | varchar(20) | 花园位置 |  |
| 6 | rating | float | 花园评分 |  |
| 7 | is\_public | boolean | 是否开放 |  |

**表 14 花园维护(Garden\_Maintenance)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | garden\_id | char(10) | 花园ID | PK FK, Garden(garden\_id) |
| 2 | maintenance\_id | char(10) | 维修ID |  |
| 3 | maintenance\_time | datetime | 维修时间 |  |
| 4 | maintainer\_id | char(10) | 维护者ID | PK FK, Account(account\_id) |
| 5 | maintenance\_description | text | 维护描述 |  |

**表 15 花园活动(Garden\_Activity)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | activity\_id | char(10) | 活动ID | PK |
| 2 | garden\_id | char(10) | 花园ID | FK, Garden(garden\_id) |
| 3 | activity\_name | varchar(50) | 活动名称 |  |
| 4 | activity\_time | datetime | 活动时间 |  |
| 5 | activity\_theme | varchar(50) | 活动主题 |  |

**表 16 花园邀请(Garden\_Invitation)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | invitation\_id | char(10) | 邀请ID | PK |
| 2 | activity\_id | char(10) | 活动ID | FK, GardenActivity(activity\_id) |
| 3 | invitee\_id | char(10) | 受邀者ID | FK, Account(account\_id) |

**表 17 商城(Points\_Mall)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | item\_id | char(10) | 商品ID | PK |
| 2 | item\_name | varchar(20) | 商品名 |  |
| 3 | item\_type | varchar(20) | 商品类型 |  |
| 4 | points\_required | int | 所需积分 |  |
| 5 | item\_storage | text | 库存 |  |

**表 18 兑换记录(Points\_Redeem)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | redeem\_id | char(10) | 兑换ID | PK |
| 2 | account\_id | char(10) | 账户ID | FK, Account(account\_id) |
| 3 | item\_id | char(10) | 商品ID | FK, Points\_Mall(item\_id) |
| 4 | redeem\_time | datetime | 兑换时间 |  |

**表 19 志愿活动(Voluntary\_Activities)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | act\_id | char(10) | 志愿ID | PK |
| 2 | garden\_id | char(10) | 花园ID | FK, Garden(garden\_id) |
| 3 | act\_description | text | 活动描述 |  |
| 4 | act\_points | int | 活动积分 |  |
| 5 | act\_recruitment\_id | char(10) | 招募ID | FK,Act\_Recruitment(act\_recruitment\_id) |
| 6 | act\_participants\_id | char(10) | 志愿者ID |  |

**表 20 志愿申请(Act\_Application)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | act\_application\_id | char(10) | 申请ID | PK |
| 2 | act\_id | char(10) | 志愿活动ID | FK,Voluntary\_Activities(act\_id) |
| 3 | account\_id | char(10) | 学生ID | FK, Account(account\_id) |
| 4 | app\_description | text | 申请描述 |  |
| 5 | application\_result | varchar(20) | 申请结果 |  |
| 6 | is\_approved | boolean | 是否通过 |  |

**表 21 志愿招募(Act\_Recruitment)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | act\_recruitment\_id | char(10) | 活动招募ID | PK |
| 2 | garden\_id | char(10) | 花园ID | FK, Garden(garden\_id) |
| 3 | recruiter\_id | char(10) | 招募者ID | FK, Account(account\_id) |
| 4 | recruiter\_time | datetime | 招募时间 |  |
| 5 | recruiter\_content | varchar(20) | 招募内容 |  |

**表 22 工作记录(Act\_Works)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段说明 | 备注（说明PK，FK等） |
| 1 | act\_id | char(10) | 志愿活动ID | FK,Voluntary\_Activities(act\_id) |
| 2 | act\_works\_id | char(10) | 工作ID | PK |
| 3 | worker\_id | char(10) | 参与者ID | FK, Account(account\_id) |
| 4 | work\_description | varchar(20) | 工作概述 |  |
| 5 | rewards | int | 积分奖励 |  |
| 6 | work\_time | datetime | 工作时间 |  |
| 7 | submission\_time | datetime | 提交时间 |  |

对应的关系模式如下：

**关系模式**

·Account(account\_id, stu\_id, password, account\_type, create\_time, login\_info\_id)

Primary key: account\_id

Foreign key: stu\_id references Student(stu\_id)

login\_info\_id references Login\_Info(login\_info\_id)

//账户

·Student(stu\_id, name, account\_id, point)

Primary key:stu\_id

Foreign key:account\_id references Account(account\_id)

//对应用户

·Administer(admin\_id, admin\_name, account\_id)

Primary key:admin\_id

Foreign key:account\_id references Account(account\_id)

//对应管理员

·Login\_Info(login\_info\_id, account\_id, login\_time, login\_location)

Primary key: login\_info\_id

Foreign key: account\_id references Account(account\_id)

//登录信息

·Violation\_Info(account\_id, violation\_description, violation\_time)

Primary key: account\_id

Foreign key:account\_id references Account(account\_id)

//违规信息，对应后面的report关系模式

·Account\_Profile(account\_id, avatar, otherinfo,registration\_time)

Primary key: account\_id

Foreign key: account\_id references Account(account\_id)

//对应账户概要

·Blog(blog\_id, account\_id, blog\_type, title, release\_time, interactive\_rating, is\_legal)

Primary key:blog\_id

Foreign key:account\_id references Account(account\_id)

//博客模块；interactive\_rating是一个需要定期更新的属性

·Blog\_Ratings(blog\_rating\_id, blog\_id, account\_id,rating\_value, rating\_time)

Primary key:blog\_rating\_id

Foreign key:blog\_id references Blog(blog\_id)

account\_id references Account(account\_id)

//博客文章评分表

·Blog\_Comment(comment\_id, blog\_id, comment\_content, commentator\_id, comment\_time)

Primary key: comment\_id, blog\_id

Foreign key: blog\_id references Blog(blog\_id)

commentator\_id references Account(account\_id)

//博客评论，通过blog\_id关联

·Apply(apply\_id, account\_id, apply\_type, apply\_time, is\_approved)

Primary key: apply\_id

Foreign key: account\_id references Account(account\_id)

//申请

·Report(report\_id,reporter\_id,reported\_user\_id,

reported\_content\_id, report\_type, report\_time)

Primary key: report\_id

Foreign key: reporter\_id references Account(account\_id)

reported\_user\_id references Account(account\_id)

//用户举报记录表

·Report\_approval\_history(approval\_id,report\_id, approval\_result, approval\_time)

Primary key: approval\_id

Foreign key: report\_id references Report(report\_id)

//举报审核记录表；包含各类举报；不仅限于blog和garden

·Garden(garden\_id, owner\_id, create\_time, garden\_pos, garden\_size, rating, is\_public)

Primary key:garden\_id

Foreign key:owner\_id references Account(account\_id)

//花园信息

·Garden\_Maintenance(garden\_id,maintenance\_id, maintenance\_time,

maintainer\_id, maintenance\_description)

Primary key:garden\_id,maintenance\_id

Foreign key:garden\_id references Garden(garden\_id)

maintainer\_id references Account(account\_id)

//花园维护信息 此表对garden做信息补充

·Garden\_Activity(activity\_id,garden\_id, activity\_name, activity\_time, activity\_theme)

Primary key:activity\_id

Foreign key:garden\_id references Garden(garden\_id)

//花园活动表

·Garden\_Invitation(invitation\_id, activity\_id,invitee\_id)

Primary key: invitation\_id

Foreign key: activity\_id references Garden\_Activity (activity\_id)

invitee\_id references Account(account\_id)

//活动邀请

·Points\_Mall(item\_id, item\_name, item\_type, points\_required, item\_storage)

Primary key:item\_id

//商城内容

·Points\_Redeem(redeem\_id,account\_id,item\_id, redeem\_time)

Primary key:redeem\_id

Foreign key: account\_id references Account(account\_id)

item\_id references Points\_Mall(item\_id)

//兑换记录

·Voluntary\_Activities(act\_id, garden\_id, act\_description, act\_points, act\_recruitment\_id, act\_participants\_id)

Primary key: act\_id

Foreign key: act\_recruitment\_id references Act\_Recruitment(act\_recruitment\_id)

garden\_id references Garden(garden\_id)

//志愿活动信息

·Act\_Application(act\_application\_id, act\_id, account\_id, app\_description, application\_result,is\_approved)

Primary key: act\_application \_id

Foreign key: account\_id references Account(account\_id)

act\_id references Voluntary\_Activities(act\_id)

//申请参加志愿活动信息

·Act\_Recruitment (act\_recruitment\_id, garden\_id,recruiter\_id,recruiter\_time,recruiter\_content)

Primary key: act\_recruitment\_id

Foreign key: garden\_id references Garden(garden\_id)

recruiter\_id references Account(account\_id)

//志愿活动招募信息

·Act\_Works (act\_id, act\_works\_id, worker\_id, work\_description, work\_time, submission\_time, rewards)

Primary key: act\_works\_id

Foreign key: act\_id references Voluntary\_Activities(act\_id)

worker\_id references Account(account\_id)

//工作记录

1. 图表索引

图 [总用例图 1 4](#_Toc135572021)

[图 2-1 系统架构设计 1 6](#_Toc135914260)

[表格 2-2 登录和注册功能的动作序列 1 5](#_Toc135572035)

[图 2-2-1 登录和注册子系统鲁棒性 1 8](#_Toc135914405)

[图 2-2-2 登录和注册功能实现 1 6](#_Toc135572133)

[表格 2-3-1 交互功能的动作序列 1 7](#_Toc135572042)

[表格 2-3-2 举报功能的动作序列 1 8](#_Toc135572051)

[图 2-3-1 鲁棒性—论坛子系统1 11](#_Toc135914535)

[图 2-3-2 鲁棒性—举报功能 1 12](#_Toc135914517)

[图 2-3-3 论坛功能实现 1 9](#_Toc135572194)

[图 2-3-4 类图—评论 1 14](#_Toc135914654)

[表格 2-4 维护功能的动作序列 1 10](#_Toc135572066)

[图 2-4-1鲁棒性—查看花园 1 16](#_Toc135914676)

[图 2-4-2 鲁棒性—修改花园 1 16](#_Toc135914686)

[图 2-4-3 维护功能实现 1 11](#_Toc135572202)

[图 2-4-4 类图—修改花园 1 17](#_Toc135914732)

[图 3 ER图 1 14](#_Toc135572226)

[总体关系表 1 15](#_Toc135572230)