



毕业设计（论文）中期报告

题目： 社区食堂系统的设计与实现

院（系） 计算机科学与技术学院

专 业 软件工程

班 级 20060213

姓 名 张宇涛

学 号 2020032951

导 师 陈芳

2024 年 3 月 18 日

撰写内容要求（可加页）：

1. 设计（论文）进展状况

(1) 理论进展状况

1) MVC 技术

系统的架构设计采用 MVC 模式，它具备高内聚、低耦合的特性。MVC 可以进行应用程序分层开发。它通过分离业务逻辑、数据与界面来编写代码，将大量业

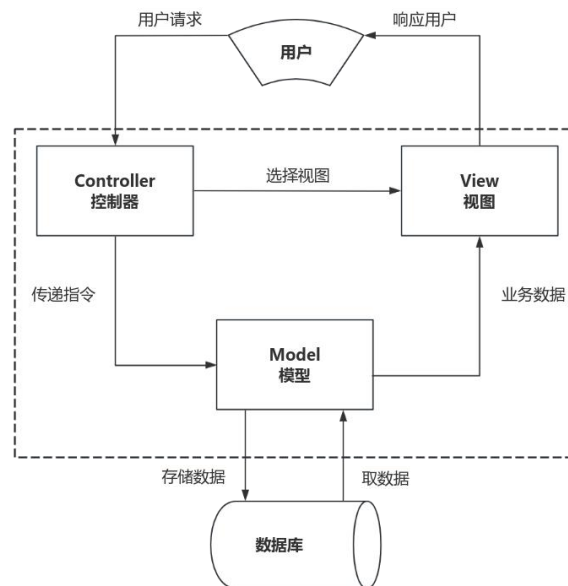


图 1.1 MVC 架构

务逻辑收集到单个组件中，在改进界面及用户交互的同时，不必重写业务逻辑，从而减少了开发者编码的时间，提高了工作效率，团队分工更明确，提升了开发代码复用性和可维护性。

如图 1.1 所示，在 MVC 设计中，模型是对数据和数据操作进行封装，并且实现数据逻辑处理。视图用于可视化模型数据并与用户交互，同时向控制器发送用户输入数据。控制器首先接收前台用户请求，再确定适当的模型处理请求，最后决定调用相应视图对返回的数据进行显示，它使视图与模型分离。

2) 信号槽技术

系统使用信号槽技术来实现社区食堂系统中多种功能之间的相互交互。Qt 中的信号槽（Signal-Slot）机制是一种用于处理对象间通信的重要技术，它基于信号和槽的概念。

信号（Signal）：信号是一种在特定事件发生时由 QObject 派生类发出的通

知。信号可以带有参数，用于向槽传递信息。在 Qt 中，信号通常通过关键字 `signals` 声明，并由特定事件触发时发出。

槽 (Slot)：槽是一个特殊的成员函数，用于响应信号。当连接的信号被发出时，与之关联的槽函数将被调用。槽函数可以处理信号传递的参数，执行相应的操作或处理逻辑。

信号槽连接 (Signal-Slot Connection)：信号槽连接是指将信号与槽关联起来的过程。在 Qt 中，可以使用 `connect` 函数将一个信号连接到一个槽，从而建立二者之间的关联。

信号映射 (Signal Mapping)：信号映射是一种高级的信号槽机制，允许将一个信号映射到多个槽。这对于一对多的事件处理非常有用。

自定义信号槽：在 Qt 中，也可以自定义信号和槽，以实现特定的对象间通信需求。通过自定义信号槽，开发者可以实现更灵活和定制化的对象间通信方式。

信号槽机制是 Qt 框架中非常重要的一部分，它提供了一种松耦合的方式，使得不同对象能够在程序中进行有效的通信和交互。这种机制使得 Qt 开发变得灵活、可扩展，并且更容易维护与调试。

3) 事件处理技术

系统通过事件处理技术实现社区食堂系统与用户之间的隐藏交互功能。QT 中的事件处理技术是指处理用户交互事件和系统事件的机制。

事件 (Event)：事件是在应用程序中发生的特定行为或状态变化，如鼠标点击、键盘输入等。在 Qt 中，事件被封装为 `QEvent` 的子类，每种类型的事件对应着不同的事件类。

事件接收器 (Event Receiver)：事件接收器是能够接收事件并进行处理的对象。在 Qt 中，几乎所有的用户界面元素都是事件接收器，能够接收和处理与其相关的事件。

事件过滤器 (Event Filter)：事件过滤器是一种用于截获和处理特定事件的机制。通过安装事件过滤器，开发者可以对任何 `QObject` 对象的事件进行监听和处理。

事件处理函数 (Event Handling)：事件处理函数是指对象中用于处理特定事件的函数。在 Qt 中，常见的事件处理函数包括 `event()` 函数和特定事件类型的虚拟函数（如 `mousePressEvent()`、`keyPressEvent()` 等）。

事件分发 (Event Dispatching)：事件分发是指将事件传递给正确的事件

接收器的过程。在 Qt 中，事件分发由事件派发器（event dispatcher）来负责，确保事件被正确地传递和处理。

自定义事件（Custom Events）：在 Qt 中，开发者还可以创建自定义事件，并通过事件派发机制，使得这些事件能够在应用程序中传递和触发。

通过使用 Qt 的事件处理技术，开发者可以轻松地实现对用户输入和系统事件的处理，从而实现交互性强的应用。同时，Qt 的事件处理机制也使得开发者能够更加灵活地定制应用程序的行为。

4) 轮播图技术

系统中的重要公告和通知采用轮播图的方式轮播展示，其主要优点如下：

吸引注意力：轮播图可以吸引访问者的注意力，引导他们注意到网站或应用中的关键信息和功能。

节省空间：通过在有限的空间内展示多个内容，轮播图有利于节省页面空间，同时确保多个信息得到展示。

多样性：轮播图允许展示多种不同内容，例如产品信息、推广活动、特别优惠等，从而增加用户与品牌的互动。

提升用户体验：优秀的轮播图设计能够提升用户体验，使页面更具吸引力和交互性。

支持响应式设计：轮播图可以很好地支持响应式设计，不论是在桌面端还是移动设备上，都能提供良好的展示效果。

突出特色内容：利用轮播图技术，重要的通告、特色产品或服务等可以在首页突出展示，提高用户认知度。

交互性：许多轮播图组件支持交互功能，如点击、滑动等，从而鼓励用户进行更多的互动和探索。

实时更新：轮播图可以随时更新内容，使网站或应用能够及时传达最新信息。

品牌展示：通过高质量的轮播图设计，可以有效展示品牌形象，增强品牌认知度。

5) 数据库技术

使用数据库而非文本形式存储数据的优点在于：

数据集中性：数据库能够将数据集中存储，使得数据管理更加高效，避免了数据分散导致的混乱。

数据共享：数据库技术能够支持多用户同时访问和共享数据，提高了工作效

率和协同合作能力。

数据完整性：数据库系统能够通过约束和验证机制确保数据的完整性，减少了数据的重复和错误。

数据安全：数据库提供了安全功能，例如权限控制、加密和备份机制，有助于保护数据的安全性。

数据一致性：数据库系统能够确保数据的一致性，避免了数据冲突和不一致的情况。

数据可追溯性：数据库可以记录数据变更的历史，包括谁、什么时间对数据进行了修改，提高了数据的可追踪性。

快速检索和分析：数据库系统能够快速对大量数据进行检索和分析，帮助用户更好地利用数据做出决策。

当前系统中使用的 logininfo 表结构具体内容，如图 1.2 的 E-R 图和表 1.1 结构所示，本系统数据库设计统一采用单一的 NOT NULL 约束作为所有数据存储表的约束条件，不进行更多的约束增加，其优点在于，简化底层功能，使得每层操作之间各司其职减少判断错误和数据混乱的问题，数据库中的数据只完成其底层的数据持久化存储功能，所有对于其数据有效性和数据间关联的判断都交给上层进行处理，减少数据混乱和计算复杂度。

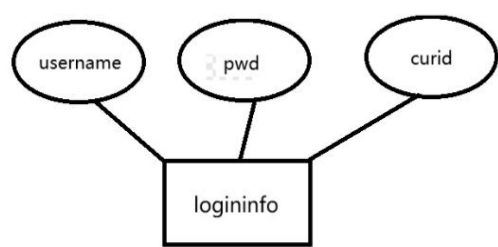


图 1.2logininfo

表 1.1logininfo 表结构

字段	数据类型	约束
username	varchar(10)	not null
pwd	varchar(20)	not null
curid	varchar(20)	not null

(2) 系统进展状况

1) 系统整体功能进展状况说明

社区食堂系统已简单完成：登录注册模块、管理员管理模块、系统公告模块，还需继续完善的功能有：在线订餐模块、订单管理模块、用户信息模块。

2) 各功能进展状况说明

登录注册界面展示

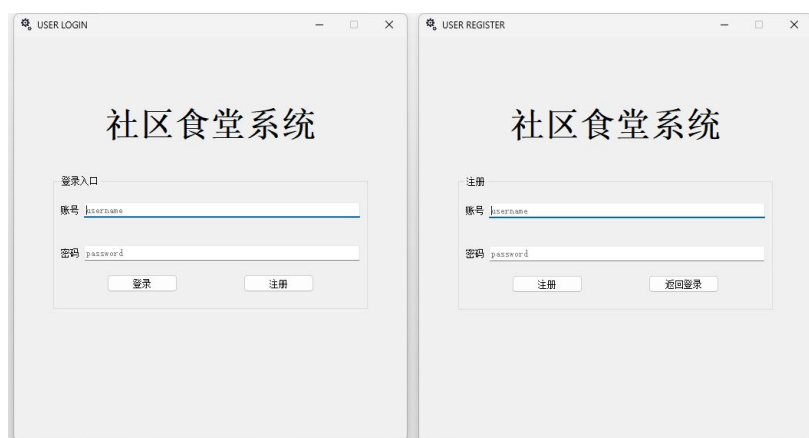


图 1.3 登陆注册界面

登录注册功能主要使用数据库技术对数据进行存储和查询，结合信号槽技术实现界面交互，事件处理技术实现对特殊按键的快速响应。

登录注册模块今后还会有更多新的功能集成其中，比如，验证码验证功能、记住密码功能、登录注册数据格式规范功能等。见图 1.3 所示。

主界面展示

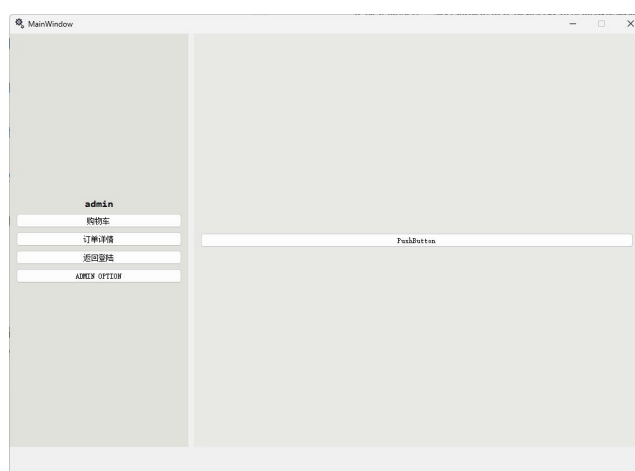


图 1.4 主界面

主界面暂时已实现管理员管理模块功能，待完善功能有：在线订餐模块、订

单管理模块和用户信息模块，暂时无法展示。见图 1.4 所示。

系统公告轮播图展示



图 1.5 系统公告轮播图

社区食堂系统公告轮播图当前已基本完成，该轮播图暂时主要为商家添加菜单公告使用，轮播图可以实现 2 秒内自动轮播系统公告，也可以实现点击左右按钮实现上一张和下一张公告信息的切换。

后续该功能可能会添加文本信息窗口界面供管理员添加系统文字公告使用，该轮播图当前实现了单一封装功能的开发，后续还需将功能嵌入系统主界面中。见图 1.5 所示。

管理员管理界面展示

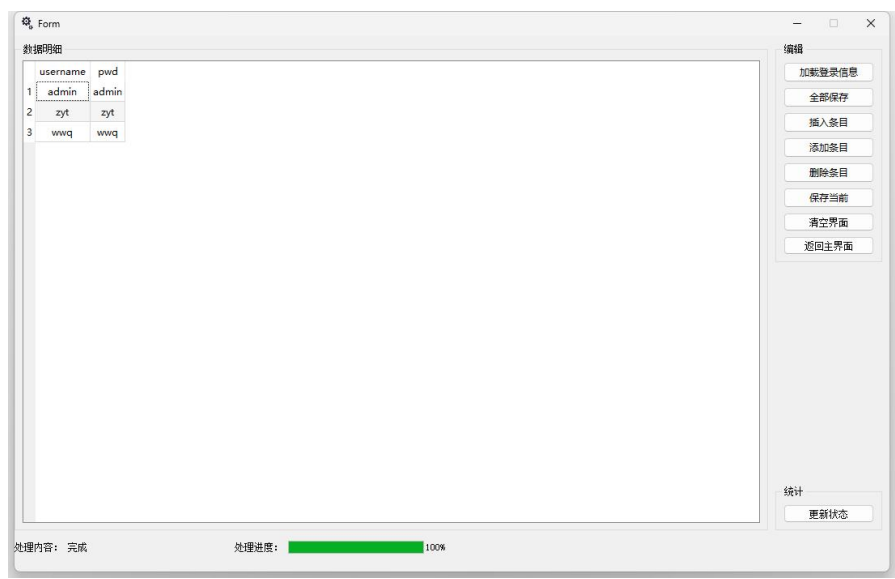


图 1.6 管理员管理界面

管理员界面中当前已完善的功能为用户登录信息管理功能，已实现对用户登

录信息的增删改查。

管理员界面模块待后续完善订餐模块、订单模块和用户信息模块后将对管理界面模块进行对应的扩展。见图 1.6 所示。

2. 存在问题及解决措施

(1) 登陆注册模块今后还会有更多新的功能集成其中，比如，验证码验证功能、记住密码功能、登录注册数据格式规范功能等，待系统主体框架全部完成后，将对这些功能做进一步优化和完善。

(2) 主界面模块待完善功能有：在线订餐模块、订单管理模块和用户信息模块，待系统主体框架全部完成后，将对这些功能做进一步优化和完善。

(3) 系统轮播图功能可能会添加文本信息窗口界面供管理员添加系统文字公告使用，该轮播图当前实现了单一封装功能的开发，后续还需将功能嵌入系统主界面中，待进一步完善系统时实现该功能。

(4) 管理员界面模块已有功能过于单一，功能进一步完善需要在后续研发中完成订餐模块、订单模块和用户信息模块后，再对管理界面模块进行进一步对应的扩展。

3. 后期工作安排

任务安排：

9-11 周：完成社区食堂系统主体主要功能框架。对各模块功能之间的交互和各功能本体进行进一步补充和完善。优化系统 UI 界面，实现更加人性化的社区食堂系统界面。

12-16 周：继续完成毕业论文，第一章：绪论。第二章：数据库和 MYSQL 技术的概念和应用。第三章：QT 技术的概念和应用。社区食堂系统完善和毕业论文第四章：社区食堂系统功能的需求分析和功能关系。第五章：社区食堂系统展示。第六章：总结。将交叉执行，互相弥补争取尽快完成论文初稿。

注：1、正文：宋体小四号字，行距 22 磅。

2、中期报告装订入毕业设计（论文）附件册。