

# PKU eater 作业报告

by 烫烫烫镊斤拷队 2200015875 曹朕豪 2300013193 张钦策 2300013090 朱兴阳

## PKU eater 功能介绍

### 简介

PKU eater 是一款以用户为中心的餐饮美食软件。旨在解决北大同学们“今天吃什么？”的终极难题。PKU eater 将提供一个同学们自由交流与分享美食的平台。在这一平台上，同学们可以自由上传菜品，发表评价，搜索最新的美食信息，或者直接让算法帮您决定今天的午餐。

### 详细功能

#### 一.菜品详情显示

用户可以通过点击主页或搜索结果页面中菜品的缩略图进入菜品详情界面，该界面将直观地展示以下信息：

- 菜品图片
- 供应窗口
- 用户评分
- 用户评价

#### 二.菜品搜索

用户可以通过搜索栏随时搜索想要的菜品，包括：

- 使用关键字搜索
- 直接选择分类进行查看（如按照供应窗口/供应时间进行查看）
- 带有分类的关键字搜索

#### 三.菜品上传

用户可以上传新菜品（即使是不完整的菜品信息），或更新菜品信息。后台数据库内将以名字作为区别菜品唯一性的标识。为保证数据的准确性，上传的菜品将由后台审核后发布。

#### 四.菜品评价

用户可以对菜品通过以下几种方式进行评价：

- 快捷评分：用户可以在菜品详情页中通过点赞/踩来对菜品进行评价,软件将综合所有用户的评价对菜品打分。
- 详细评价：用户可以在菜品详情页中通过输入评价内容来对菜品进行评价，评价内容将在菜品的评论区展示。

#### 五.美食推荐

用户可以一键获得一道美食推荐，解决“今天吃什么”的终极难题。

- 使用随机推荐算法

# PKU eater 实现细节

---

## 概述

PKU eater由两部分构成：服务器端与客户端，服务器端负责菜品的存储，搜索算法的实现。客户端负责界面的显示与切换，菜品与评论的上传。而服务器端与客户端的通信通过tcp模块实现。

## 服务器端

### 搜索算法实现(utilities.h)

搜索算法与推荐算法被整合在了一起，具体来说，utilities.h中实现了一个搜索的模板函数，通过传入的比较函数对当前数据库中的所有菜品生成一个“匹配度”评分，并返回评分靠前的若干菜品名称。

该头文件中提供了三种计算匹配度的函数:随机返回匹配度（Compare1），返回是否为子串（Compare2），返回将编辑距离归一化至0-1之间的函数（Compare3），第二种算法并未被实际应用。

当推荐按钮被点击时，随机返回匹配度的搜索算法被调用，当搜索按钮被点击时，使用编辑距离的搜索算法（并加入标签是否符合的判断）被调用。

### 通信消息约定(msg.h)

为了服务器和客户端之间通信的方便，约定互相发出的消息形式，包括服务器发出的ServerMsg，客户端搜索发出的ClientMsg和上传菜品发出的UpLoadMsg。

每个消息类型都约定了某些变量用于消息处理，具体参见注释。

### 自定义的TCP服务器(myserver.h)

实际上，Qt自带了QTcpServer类，但是并不能满足实际需要，为此从其派生出MyServer类，该类中包括了一个QVector用于储存所有连接到该客户端套接字，一个QVector用于储存所有的菜品。一个QString用于记录数据库位置。

当MyServer对象被初始化时，除了初始化监听端口外，将从数据库位置读取所有的json文件，并生成对应的Cuisine对象储存在内存中。

当有新的socket连接到MyServer时，将触发重写的incomingconnection函数，将socket的clientdisconnected信号和处理客户端断联的handle\_client\_disconnected函数连接在一起，将sendmsg信号和处理msg的函数连接在一起。

在handle\_client\_disconnected函数中，服务器将移除对应的socket，并输出结果。

在处理msg的函数中，服务器将按照约定的顺序解析各项参数，并调用合适的搜索算法，或者传输指定的二进制json文件，或者对QVector内的Cuisine对象进行点赞/点踩/评论操作。

### 自定义的TCP套接字(mysocket.h)

实际上，Qt自带了QTcpSocket类，但是并不能满足实际需要，为此从其派生出MySocket类，该类在原有基础上添加了发送json和string的函数，读取clientmsg的槽函数，客户端关闭连接的槽函数以及两个自定义的信号用于通知服务器msg准备完毕和客户端关闭连接

发送json和string的函数具有相同的结构，即将对应的对象转换为二进制数据流写入socket，区别在于string函数被设计为发送搜索算法的结果，所以在确定发送内容时会调用一次搜索函数。而json函数仅需要欲传输的json地址。

receivedata函数用于将得到的数据打包为msg，并将触发对应信号，将其传递给MyServer，而

slotclientdisconnected函数仅用于触发clientdisconnected信号，包括该MySocket自身的对应编号，以便服务器进行处理。

## 客户端

### 菜品类(cuisine.h与json\_cuisine.h)

菜品类储存了所有菜品的信息，出于程序的健壮性考量，这些信息都被设定为私有的，使用对应的get()与set()接口进行操作，同时菜品类储存了所有的评分与评论，并提供了修改评分的接口和新增评论的接口。并且菜品类提供了便捷的生成测试用例的函数。

菜品以json形式储存在本地，json\_cuisine.h提供了从json中读取cuisine的函数和将菜品储存为json文件的函数。

### 用户端主界面绘制(mainwindow.h, mainwindow.ui)

按钮、搜索框等组件的大小，位置在mainwindow的构造函数中进行设置。ui界面的绘制：

重载了QWidget::paintEvent：QT中窗口需要重绘时（例如被覆盖后重新显示），会自动触发paintevent函数的执行。在函数内部，可以通过RGB值设定Qcolor对象的颜色，然后创建Qpainter对象，利用drawRect等方法进行绘制。

通过Stylesheet设定按钮等组件的形状，颜色等性质：例如mainwindow.ui中将推荐按钮的圆角半径设置为按钮长、宽的一半，从而将按钮变为圆形

### 菜品详情界面(cuisinedetails.h, commentdetails.h)

- 1.点赞点踩：一个用户，只能点赞或者点踩一次。
- 2.菜品图片自适应拉伸，保持比例。
- 3.精选评论可以从所有短评论中精选三条（目前算法为随机）。
- 4.可以刷新精选评论。
- 5.点击查看所有评论会弹出所有评论的窗口。
- 6.使用了与主界面一致的绘制方式

### 客户端套接字(clientsocket.h)

从QTcpSocket派生出的类，主要实现了优化的read\_data函数，用于处理tcpsocket原有的传输限制问题。具体而言，每次有新数据进入缓存区触发read\_server\_msg槽函数时，先将传入的数据统一缓存在一个数据流中。首先判定是否为头文件，读取得到文件大小和类型信息。再通过已接收到的字节数判定接受是否完成。一旦完成触发msg\_ready信号，将得到的数据打包为msg传出。

为前端界面开放request接口，通过该接口请求数据，发送信息到服务器，通过msg\_ready信号对应的槽函数获取服务器传回的消息。由此实现了前后端的解耦。

### 菜品上传界面(upload.h)

窗口启用了拖拽与放下事件，当拖拽文件进入时，检测其后缀，允许png与jpg文件放入。放入文件时，读取路径并读入图片。以此实现了直观的图片上传      使用了与主界面一致的绘制方式

## 小组成员分工情况

---

在项目初期，我们通过头脑风暴的方式，以北大同学的需求为核心，提出了诸如创意文字游戏、匿名交友平台、美食测评与推荐平台等，最终确定开发美食测评与推荐平台，参考其他北大系软件，命名为PKU Eater。此阶段，小组三位同学百家争鸣、百花齐放，均提出了意义非凡的想法。

确定开发项目后，一方面我们讨论确立了PKU Eater的设计理念，将其定位为一款轻量简约、受众明确的校园APP；另一方面，我们继续运用头脑风暴的方式，提出了PKU Eater应该实现的功能，包括菜品详情、菜品搜索、上传新菜品、菜品评价、菜品推荐五大方面。小组成员对此贡献均等。

紧接着，我们确定了程序架构，将项目分割为菜品类、长评论类、工具算法、UI、服务器通讯五大程序模块；针对这些模块，我们进行了第一次分工：

- 曹朕豪负责客户端UI、tcp传输；
- 张钦策负责菜品类、菜品类转换为json存储；
- 朱兴阳负责长评论类、搜索算法。

在第一次实现过程中，曹朕豪发现tcp传输技术复杂、实现困难，小组决定由他专攻tcp通讯，而张钦策、朱兴阳临危受命，分别负责菜品搜索界面和菜品详情界面的UI设计。小组成员分别独立在各自设备上构建测试环境，并且完整实现了负责的功能。

在各个成员完工后，本小组开展了对接工作，在曹朕豪的传输接口下，运行各自的窗口并进行了Debug。经过一整天的奋战，我们最终完成了设计中的绝大部分功能。在对接阶段，小组成员分别为负责的功能Debug。

- 最后，在功能基本实现的情况下，我们制作了功能演示PPT，录制了展示视频。其中：
- 曹朕豪负责功能演示PPT、操作软件；
- 张钦策、朱兴阳负责展示视频的录音。

# 总结

---

## 团队合作

在项目初期，我们通过头脑风暴提出了多个创意，并最终确定了开发PKU Eater。这一过程增强了我们团队的合作和沟通能力。每个成员都积极参与讨论，贡献了自己的想法，最终形成了统一的设计理念。

## 自主学习

项目开发过程中，我们分别负责不同的模块，深入学习了相关技术。比如，曹朕豪专注于TCP传输的实现，张钦策负责菜品类及其JSON存储，朱兴阳则负责长评论类和搜索算法的实现。通过解决实际问题，我们在各自负责的领域得到了深入的理解和提升。

## 抗逆能力

在开发过程中，我们遇到了一些技术难题，尤其是在TCP传输方面。通过团队的共同努力和调整分工，我们最终克服了这些困难。比如，曹朕豪专攻TCP通讯，而张钦策和朱兴阳则临时负责菜品搜索和详情界面的UI设计。这样的调整展现了我们团队的灵活性和应变能力。

## 统筹能力

各个模块开发完成后，我们进行了集成和调试。通过一整天的奋战，我们实现了设计中的绝大部分功能，并确保了系统的稳定性和可靠性。

## 未来展望

虽然我们已经实现了预期的大部分功能，但仍有一些细节需要完善。我们计划在后续的开发中进一步优化用户体验，增加更多实用功能，提升系统的稳定性和速度。总的来说，这次项目不仅让我们在技术上有了新的突破，也让我们体会到了团队合作的重要性，提升了解决问题的能力。我们期待在未来的项目中能够继续发挥所学，不断进步。