**聊天软件概要设计文档**

**1.引言**

**1.1编写目的**

根据《软件综合开发实践》课程第七章节需求分析，得出了系统的基本需求，要实现整个系统，需要对教学大纲的要求进行设计，概要设计主要是利用比较抽象的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。预期读者为综合开发组教师人员。

**1.2项目背景**

根据教学大纲要求，第七章节实践训练课程需要将所学知识融汇贯通，并提高学生的动手能力，就给教师团队提出了要求：提供实践课程的教案、ppt和相应程序代码。为了更好的串联知识点，经小组讨论，提出用聊天软件的开发来进行工作的展开。

|  |  |
| --- | --- |
| 系统名称： | 聊天软件系统 |
| 任务提出者： | 惠普(济宁)软件综合开发课程小组 |
| 开发者： | 本小组 |
| 用户： | 全体教师团队、全体学生 |

**1.3定义**

* GTK：界面开发库
* C/C++：软件开发语言
* SOCKET：网络通信协议
* MySQL：数据库

**1.4参考资料**

《软件综合开发实践》蔡建平编著 惠普内部资料

《GTK+2.0编程范例》宋国伟编著 清华大学出版社

**2.任务概述**

**2.1目标**

完成聊天软件client端与server端的研发

**2.2运行环境**

虚拟机 CentOS操作系统

**2.3需求概述**

通过该系统的实施,使得GTK编程、SOCKET通信技术、MySQL数据库有效结合。按照需求：

1. 聊天client端能相互通信，聊天数据能及时发送给server端，client端能查看聊天历史记录，能修改签名。
2. 聊天server端提供操作MySQL数据库的socket接口，管理员登陆验证，初始化数据库，管理client端信息，处理客户端请求，server端同步上下线信息，server端查看聊天记录，查看日志。

**3.总体设计**

**3.1处理流程**

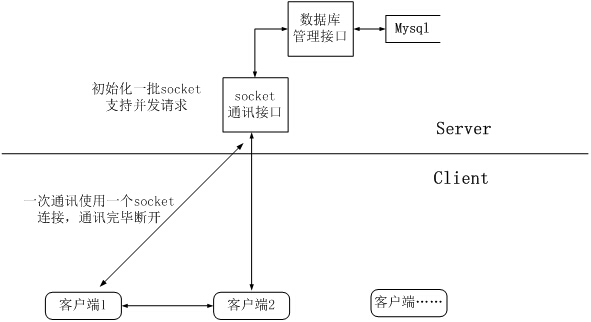


图3.1 处理流程

**3.2总体结构和模块外部设计**

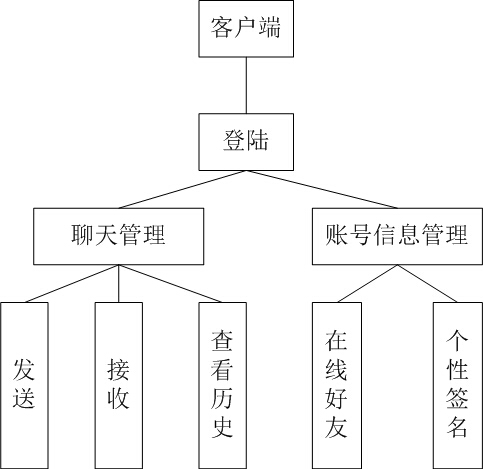


图3.2 客户端结构模块



图3.3服务端结构模块

**3.3功能分配**

客户端：发送聊天信息，接收聊天信息，查询聊天历史，登录，查询在线人员，修改个性签名。

服务端：管理员登陆，数据库初始化，客户端请求处理，客户端信息管理，聊天记录管理。

**3.4人工处理过程**

客户端：IP配置、发送聊天内容、查询聊天记录

服务端：查看客户端信息，查看聊天记录，查看日志

**3.5尚未解决的问题**

客户端：多客户端并发聊天，文件传输

服务端：数据库相关操作的界面实现

**4.接口设计**

**4.1外部接口**

用户界面部分，根据需求分析结果，用户需要一个用户友善界面。考虑到章节知识，界面用GTK库，使用C\C++语言开发。

**4.2内部接口**

客户端：登录模块返回值决定当前用户是否可以使用聊天客户端，登录成功后根据各项子功能模块进行处理。发送模块根据录入内容向相应IP发送信息，并向server发送同样信息；接收模块被动接收信息，并显示内容；查看历史记录模块向server发送请求，接收内容并显示；添加好友模块和修改备注模块向server发送请求。

服务端：管理员登陆，与数据库表对比验证；接收客户端上下线、修改签名请求，返回结果并修改数据库表；接收客户端查看历史请求，读取数据库表中内容，向客户端返回历史记录；查看客户端信息、查看聊天记录，读取数据库表中对应内容进行显示。

**5.数据结构设计**

前期为满足server端数据表格创建，特进行相应数据结构设计。

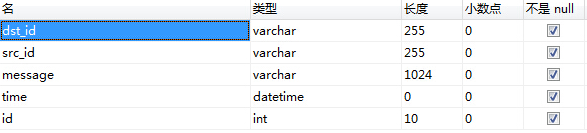
**5.1逻辑结构设计要点**

server端默认为mysql，相应表格设计如下：

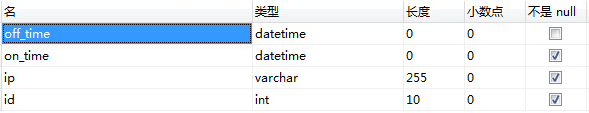
user用户表



chat\_20140815聊天记录表

****

log\_20140815用户上下线表

****

**5.2物理结构设计要点**

物理数据结构设计主要是设计数据在模块中的表示形式。数据在模块中都是以结构的方式表示。用户信息和聊天信息参照上表。

**5.3数据结构与程序的关系**

程序在终端对信息进行操作时需要间接对数据库数据结构进行操作，也就是以上表格：登录验证，聊天记录存储，历史记录查看等都需要对相应表格进行间接操作。

**6.运行设计**

**6.1运行控制**

运行控制将严格按照各模块间函数调用关系来实现。

客户端：

1. 在socket通信操作方面，client端发送到server端数据后，登录验证和查看历史模块需要等待server端返回数据，client端根据返回结果显示操作结果或错误提示；
2. 发送聊天信息和个性签名模块只是向server发送信息；
3. 在线好友模块被动等待server端发送信息，client端根据接收信息显示好友信息。

服务端：

1. server端被动等待client端上下线请求，读取、修改数据库，返回操作结果或错误提示；
2. server端接收client端修改个性签名请求，修改数据库对应项，返回操作结果或错误提示；
3. server端接收client端查看历史请求，读取数据库对应内容，返回数据或结果；
4. server端接收客户端发送信息内容，修改数据库对应项；
5. server端定时与在线client端握手，以确定是否真正在线；
6. server端在client端处理上下线后，广播当前在线客户端列表；
7. server端主动查看客户端信息、日志、历史记录，读取数据库对应项。

**6.2运行时间**

由于本软件只是针对教学用，暂时不计较性能方面的问题，故对软件各个响应时间没有具体要求。