

# 软件设计文档

## 项目文件结构

```
- MiRoad/  
  -extend_jar/  
    -mail.jar  
    -mysql-connector-java-5.0.5-bin.jar  
  -images/  
    -img1  
    -img2  
    ...  
  -src/  
    -Constant.java  
    -ImageProcess.java  
    ...  
  -MiRoadClient.jar  
  -MiRoadServer.jar  
  -readme.txt
```

## 模块划分

1. 即时聊天模块
2. 文件/图片传输模块
3. 邮件代理模块
4. 好友添加模块
5. 心跳信息模块
6. 登陆注册模块

## 技术选型理由

1. 选择 Java 语言编程,应用 socket 编程的技术可以比较容易的实现网络通信协议,而且 Java 语言同时对于 GUI 界面的设计更有优势。
2. MySQL 作为数据库,使用关系型数据库在进行数据保存的时候,逻辑上更容易进行设计,而且 Java 提供了与数据库进行交互的包,同时 MySQL 也是目前使用最广的数据库。
3. 采用 MVC 模型进行设计,将用户之间的操作通过 Controller 进行管理,符合现代软件设计的习惯,方便管理服务器。

## 软件使用的设计技术

### OO programming && Adapter Class/Object

所在位置:

```
1+ import java.awt.Graphics;
L1
L2 // 按压效果按钮类
L3 class MyButton extends JButton {
L4     String show, in, down; // 一个按钮的3中状态图片路径
L5
L6+ public MyButton(String show, String in, String down) {
L7     super(); // 调用父类构造
L8     setContentAreaFilled(false); // 不画按钮背景
L9     setBorder(null); // 不画边框
L10
L11     this.show = show;
L12     this.in = in;
L13     this.down = down;
L14 }
L15
L16+ public void paintComponent(Graphics g2) {
L17     Image img1 = new ImageIcon(show).getImage(); // 读取本地图片
L18     Image img2 = new ImageIcon(in).getImage(); // 读取本地图片
L19     Image img3 = new ImageIcon(down).getImage(); // 读取本地图片
L20     MediaTracker mt = new MediaTracker(this); // 为此按钮添加媒体跟踪器
L21     mt.addImage(img1, 0); // 在跟踪器添加图片，下标为0
L22     try {
```

### Mediator

所在位置:

```
import java.io.IOException;

public class RunMiRoad {
    public static void main(String[] args) throws IOException {

        Myclient client = new Myclient();
        MiRoad win = new MiRoad("MiRoad", "images/MiRoadBackGround.png", 700, 620, "images/6.jpg");
        LogMiRoad LMR = new LogMiRoad("images/loginbackground.png", "images/loginheader.png");

        LMR.setMiRoad(win);
        LMR.setClient(client);
        win.setClient(client);
        client.setMiRoad(win);
        client.setLogMiRoad(LMR);
    }
}
```

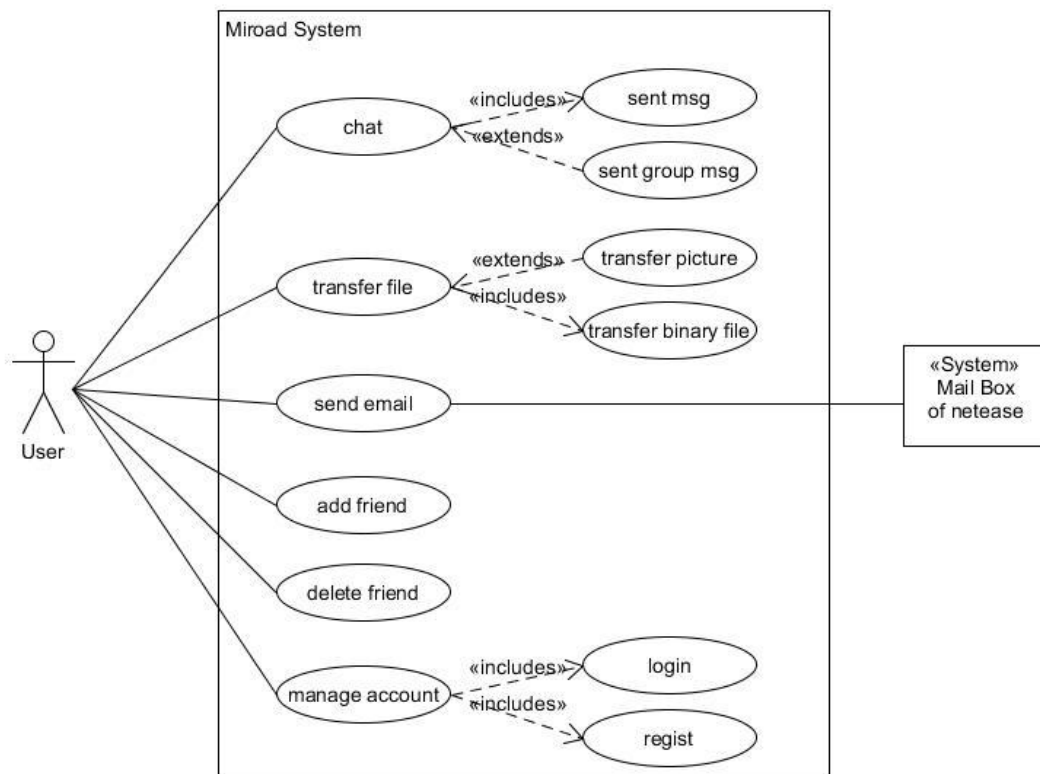
## MVC

架构视图:

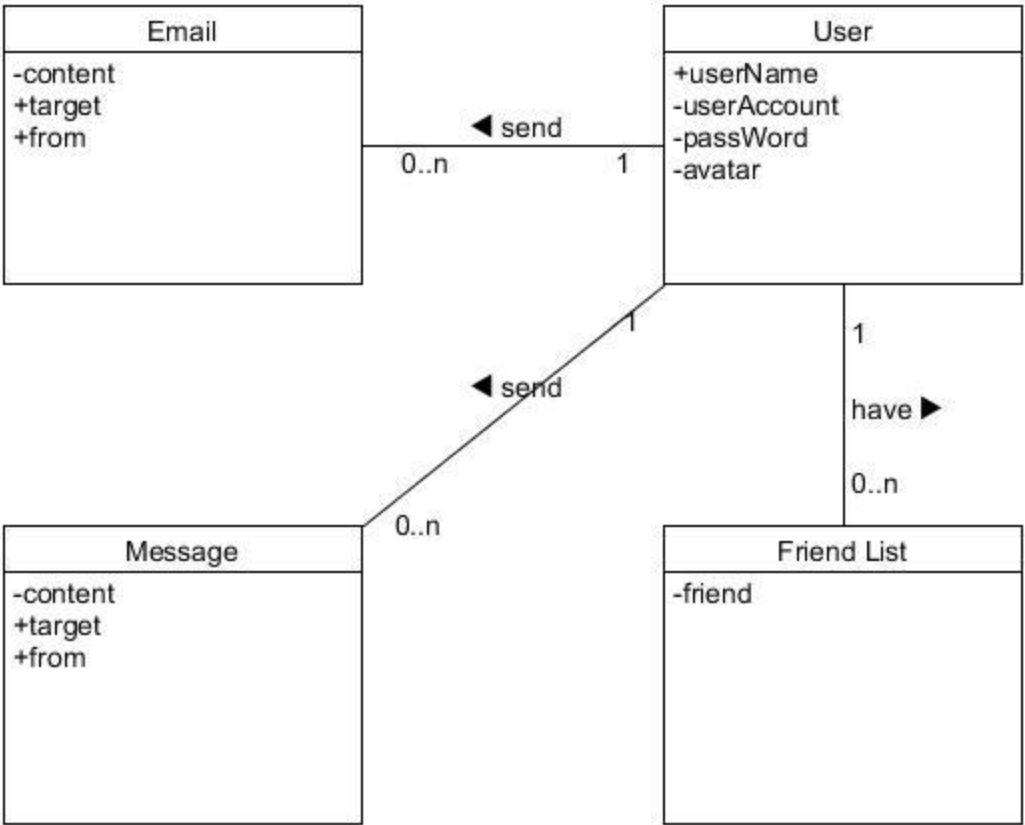
- Constant.java
  - ImageProcess.java
  - LogMiRoad.java
  - MessageProcessor.java
  - MessageProcessor2.java
  - MiMail.java
  - MiRoad.java
  - MyButton.java
  - Myclient.java
  - RunMiRoad.java
  - SendMail.java
  - Server.java
- images

## 架构设计与设计图

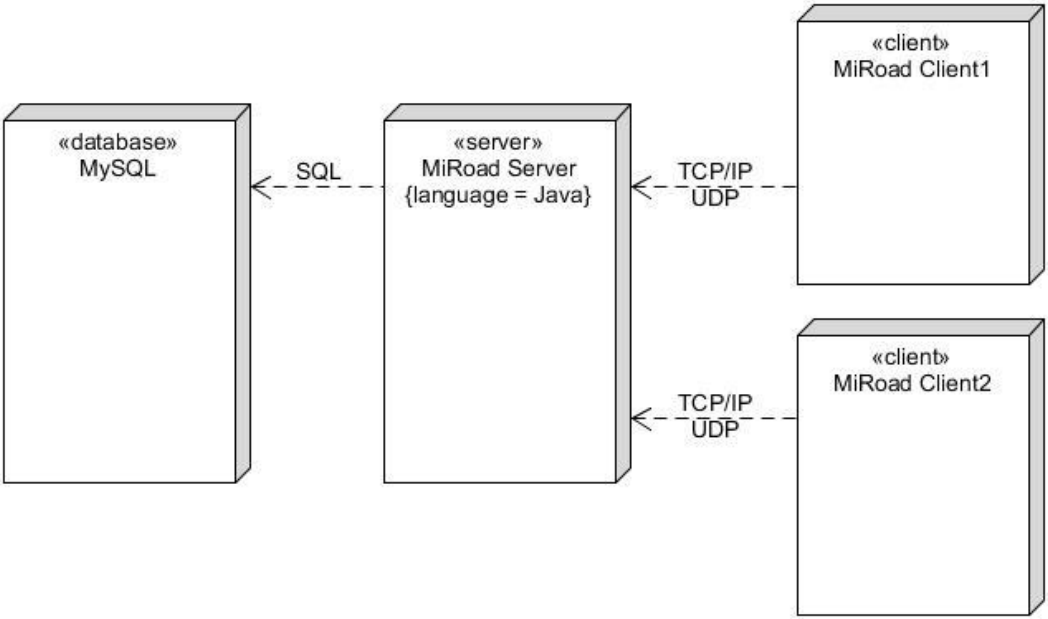
核心用例的用例图



领域模型：



架构图（逻辑部署）：



后台设计图：

