**Live-Chat在线聊天系统**

**需求规格说明书**

版本1.6

**项目组成员：**

**詹桂发**

**袁梓民**

**何璇**

**张旭唐**

**闫琰**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订号** | **描述** | **编制者** |
| **3.29** | **V1.0** | **问题陈述 需求分析3.1 3.2** | **全体组员** |
| **4.6** | **V1.1** | **规范内容 需求分析3.3 3.4** | **全体组员** |
| **4.13** | **V1.2** | **系统框架 关键抽象** | **全体组员** |
| **4.28** | **V1.3** | **用例分析 总类图** | **全体组员** |
| **5.3** | **V1.4** | **用例顺序图分析 分析类机制** | **全体组员** |
| **5.11** | **V1.5** | **分包分子系统** | **全体组员** |
| **5.27** | **V1.6** | **部署图和构件图** | **全体组员** |

目录

[第一章 引言 - 2 -](#_Toc420503380)

[1.1 文档用途 - 2 -](#_Toc420503381)

[1.2 阅读对象 - 2 -](#_Toc420503382)

[第二章 问题陈述 - 3 -](#_Toc420503383)

[2.1 背景 - 3 -](#_Toc420503384)

[2.2 用户 - 3 -](#_Toc420503385)

[2.3 功能概述 - 3 -](#_Toc420503386)

[第三章 需求分析 - 4 -](#_Toc420503387)

[3.1 用例析取 - 4 -](#_Toc420503388)

[3.2 用例规约 - 4 -](#_Toc420503389)

[3.2.2 用户获取聊天记录 - 5 -](#_Toc420503390)

[3.2.3 用户注册 - 6 -](#_Toc420503391)

[3.2.4 修改个人信息 - 7 -](#_Toc420503392)

[3.2.5 其他用例规约 - 8 -](#_Toc420503393)

[3.3 补充规约 - 8 -](#_Toc420503394)

[3.4 术语表 - 8 -](#_Toc420503395)

[第四章 Live-Chat在线聊天系统设计 - 10 -](#_Toc420503396)

[4.1 Live-Chat在线聊天系统框架 - 10 -](#_Toc420503397)

[4.2 系统关键抽象 - 10 -](#_Toc420503398)

[4.3 用例分析 - 11 -](#_Toc420503399)

[4.3.1 登录用例的类分析和设计 - 11 -](#_Toc420503400)

[4.3.2 获取用户聊天记录用例的类分析和设计 - 12 -](#_Toc420503401)

[4.3.3 修改个人信息用例的类分析和设计 - 12 -](#_Toc420503402)

[4.4 总类图 - 13 -](#_Toc420503403)

[4.5 Live-Chat 用例顺序图分析 - 13 -](#_Toc420503404)

[4.6 Live-Chat 分析类机制 - 14 -](#_Toc420503405)

[4.7 子系统设计 - 14 -](#_Toc420503406)

[4.7.1 子系统划分 - 14 -](#_Toc420503407)

[4.7.2 用户管理子系统 - 15 -](#_Toc420503408)

[4.7.3 分析类到设计元素映射 - 15 -](#_Toc420503409)

[4.7.4 设计元素及其包 - 15 -](#_Toc420503410)

[4.8 部署图 - 16 -](#_Toc420503411)

[4.9 构件图 - 16 -](#_Toc420503412)

# 第一章 引言

## 1.1 文档用途

本文档的编写目的是方便读者对Live-Chat在线聊天系统的需求和设计进行了解。

## 1.2 阅读对象

此文档适合以下人员阅读：

·本项目组员

·有兴趣了解本系统的读者

# 第二章 问题陈述

## 2.1 背景

基于浏览器的B/S系统应用正由于其便捷的使用方式越来越多地受到广大开应用开发者和用户的喜爱，而社交软件正是当前APP开发中的热门选择，然而像WebQQ这样的工具并没有很好地体现浏览器的快捷高效，因此，我们小组希望能通过自己的体验改善并设计出一款具有更高用户体验的在线聊天系统——Live-Chat。

## 2.2 用户

Live-Chat是一个非常容易上手的在线聊天系统，因此我们面向的是所有希望通过WebAPP进行社交活动的计算机使用者。

## 2.3 功能概述

A．注册/登陆/登出功能，其中登陆这边可以使用第三方账号登陆，如：微博、Github等；

B. 登陆成功后，用户可以选择修改自己的一些基本资料，如：头像等；

C．当然最重要的还是聊天功能。

·聊天列表显示所有在该列表中用户的离线/在线状态

·提示是否有离线消息

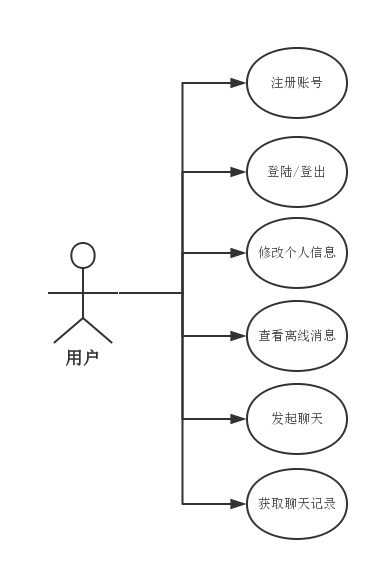
·当用户选择另一个用户进行聊天时（私聊）/或选择群聊窗口，切换到聊天窗口并显示历史记录

·两个或多个在线用户可以进行实时聊天

# 第三章 需求分析

## 3.1 用例析取

对于聊天系统而言所有的参与者都是用户。以下是本系统用例图：



## 3.2 用例规约

3.2.1 用户登录

1）简要说明

本用例描述注册用户如何登录到Live-Chat在线聊天系统。

2）参与者

注册用户

3）事件流

ⅰ.基本事件流

本用例开始于注册用户希望登录到Live-Chat在线聊天系统。

A) 系统请求注册用户输入用户名和密码；

B) 注册用户输入用户名和密码；

C) 系统验证输入的用户名和密码；

·用户名不存在

·用户名对于密码不正确

D) 注册用户成功登录到聊天界面，进行其他操作。

ⅱ.后备事件流

·用户名不存在

显示用户名不存在错误信息；

系统提示用户注册

·用户名对应密码不正确

系统显示用户名密码不正确错误信息；

返回基本事件流第一步。

4) 特殊需求

密码输入框必须以密文方式呈现，

5) 前置条件

本用例开始前注册用户已经打开对应的登陆界面。

6) 后置条件

如果用例成功，注册用户将成功登录系统，并赋予相应权限。若失败，系统状态不改变。

### 3.2.2 用户获取聊天记录

1) 简要说明

本用例描述注册用户如何在Live-Chat在线聊天系统中获取聊天记录。

2) 参与者

注册用户

3) 事件流

ⅰ.基本事件流

本用例开始于注册用户希望查看聊天记录。

1. 用户点击加载更多聊天记录；
2. 系统显示最后20条聊天记录；

·未加载聊天记录少于20条

ⅱ.后备事件流

·未加载聊天记录少于20条

系统显示剩余的全部聊天记录

加载更多聊天记录按钮显示没有聊天记录了

4) 特殊需求

无。

5) 前置条件

本用例开始前注册用户已经打开相应的聊天窗口，并且与聊天对象间存在有聊天记录。

6）后置条件

如果用例成功，注册用户的聊天窗口将加载出20条最新的聊天记录并显示是否还有聊天记录未加载。

### 3.2.3 用户注册

1) 简要说明

·本用例描述未注册用户如何在Live-Chat在线聊天系统中注册。

·用户名合法性定义：用户名必须只能包含字母和数字并且长度不超过8个字符。

·密码合法性定义：密码的长度必须超过6个字符。

2) 参与者

注册用户

3) 事件流

ⅰ.基本事件流

本用例开始于注册用户希望在Live-Chat在线聊天系统中注册。

A)系统请求注册用户输入用户名：

·用户名已存在

·用户名不合法

1. 系统请求注册用户输入两次密码：

·密码不合法

·两次密码不一样

1. 系统请求用户输入邮箱、昵称；
2. 系统将当前用户的信息增加到数据库中。

ⅱ.后备事件流

·用户名已存在

系统显示用户名已存在错误信息；

返回基本事件流第一步。

·用户名不合法

系统显示用户名不合法错误信息；

返回基本事件流第一步。

·密码不合法

系统显示密码不合法错误信息；

返回基本事件流第一步。

·两次密码不一样

系统显示两次密码不一样错误信息；

返回基本事件流第一步。

4) 特殊需求

密码输入框必须以密文方式呈现。

5) 前置条件

本用例开始前用户打开系统注册界面。

6) 后置条件

如果用例成功，用户将注册成功，系统在数据库中增加一条注册用户的相关记录。若失败，系统状态不改变。

### 3.2.4 修改个人信息

1)简要说明

本用例允许注册用户对个人信息进行管理，提供修改密码功能，但不允许修改用户名。

2) 参与者

注册用户

3) 事件流

ⅰ.基本事件流

本用例开始于注册用户希望修改个人信息。

1. 系统请求注册用户输入新密码和确认密码；
2. 注册用户输入新密码和确认密码；
3. 系统验证输入的新密码和确认密码：

·新密码不合法

·确认密码和新密码不一致

1. 系统请求注册用户输入邮箱、昵称；
2. 注册用户输入邮箱、昵称；
3. 系统修改注册用户的个人信息。

ⅱ.后备事件流

·新密码不合法

系统显示新密码不合法错误信息；

返回基本事件流第一步。

·确认密码和新密码不一致

系统显示确认密码和新密码不一致错误信息；

返回基本事件流第一步。

4) 特殊需求

密码输入框必须以密文方式呈现。

5) 前置条件

本用例开始前注册用户已经登录系统

6) 后置条件

如果用例成功，注册用户将成功修改个人信息。若失败，系统状态不改变。

### 3.2.5 其他用例规约

·登录用例规约

·离线消息用例规约

·发起私聊用例规约

·发起群聊用例规约

·查看聊天记录规约

·退出登录用例规约

## 3.3 补充规约

本部分为该系统的整体非功能性补充要求。

1. 兼容性

系统可支持所有主流浏览器。

1. 可靠性

系统每周二凌晨0点至4点需进行维护。

1. 可用性

系统使用者所需培训时间为5-10分钟，符合常见的Windows界面风格。

1. 安全性

·系统中的用户只能在被允许的范围内使用，无法干涉或修改其他用户的信息

·用户登录系统需要输入用户名和密码

1. 易用性

系统充分考虑不同使用频率功能在屏幕上的分布，给用户最佳的体验。

1. 性能

·系统可支持3000个在线用户

·加载聊天记录响应时间少于2秒

1. 设计约束

本系统使用Nodejs进行编写，可在所有主流浏览器上流畅地允许。

## 3.4 术语表

本部分内容包括与本系统开发相关的关键概念定义。

1. Live-Chat在线聊天系统

为广大用户提供社交活动平台的系统，即本次开发项目。下文中的系统均指Live-Chat在线聊天系统。

1. 用户名

注册用户在系统中与他人进行社交活动的唯一标识。

1. 注册用户

拥有系统登录权限和发起聊天活动权限的人。

1. 离线消息

指信息的接收方未登录本系统的情况下在线用户向其发送的消息。

1. 私聊

指仅发生在两个用户之间且第三个用户无法看到的聊天。

1. 群聊

指发生在两个或两个以上用户之间且系统中所有在线用户都能看到的聊天。

1. 聊天记录

系统对用户在系统中进行的对话所做的记录。

# 第四章 Live-Chat在线聊天系统设计

## 4.1 Live-Chat在线聊天系统框架

本系统基于B/S架构用Nodejs开发实现，使用三层结构进行分层，分别是界面层、业务层、数据访问层；开发选用了Nodejs的官方推荐框架Express。Express本身是一个基于MVC的框架。M代表数据层，采用非关系型的数据库MongoDB来实现数据持久化；V代表视图层，采用了Jade作为前端模板引擎请，利用模板继承和模块化等特性，实现了前端组件化；C代表控制层，主要是作为Web的路由逻辑和视图层与数据层的中介，掌控Web后端逻辑。

* 表现层

表示层负责获取用户的请求和信息的展现，是用户与系统交互的界面。表现层作为Web应用的前端部分，采取了当前流行的Web APP开发流程：

* 1. 模块化

LiveChat作为富交互的WebAPP，前端的控制逻辑先锋队复杂，为了保持前端逻辑的清晰以及保证代码的重用性，采取了模块化工具Browserify。通过Browserify，前端的JavaScript可以实现模块化编写，提高了代码的结构性。本系统表现层的模块有: 用户注册、用户登陆、聊天用户列表、聊天状态维护模块、在线用户列表、消息发送模块、消息接收模块、离线消息模块、前后端数据交互模块、数据连接模块。

* 1. 自动化工作流

为了充分利用前端流行工具的使用，采用了前端自动化工具Grunt。通过Grunt，可以自由组合前端需要的工具。本系统根据Grunt的特点，实现了具有自动化编译、自动化打包、模块化

* 业务层

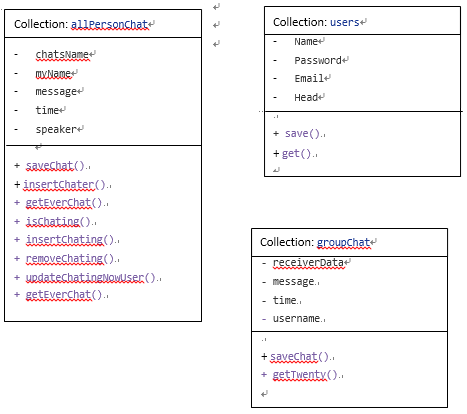
业务层负责整个WebAPP的路由以及视图层和数据层的连接。通过Node就是模块化的特点，数据层会封装成很多模块，只暴露接口，而业务层又会暴露接口给视图层，这样业务层就单纯做了中介者和逻辑控制，清晰了开发流程。

* 数据层

数据层负责系统的实体对象的数据访问。本系统采用MongoDB实现了数据持久化。MongoDB作为非关系的数据库，非常适合与Node结合开发WebApp。本系统数据层的模块包括：用户、私聊、群聊

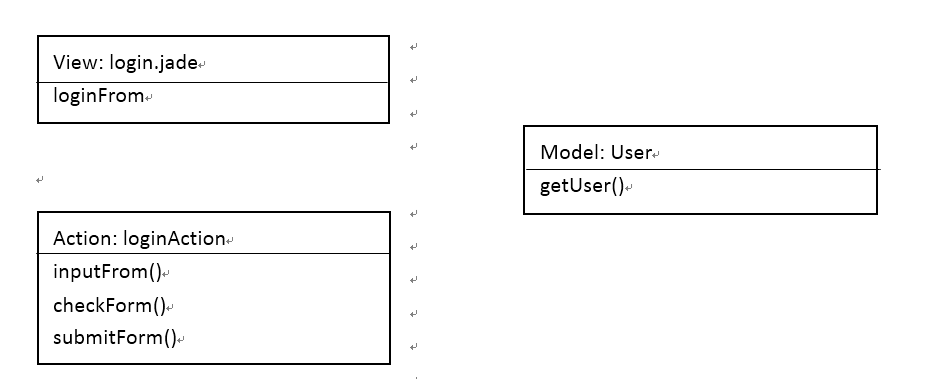
## 4.2 系统关键抽象

系统关键抽象即系统实体类图。本系统采用的是非关系型数据库，不同于传统的关系型数据库，没有表与外键等概念。每一个“表”就是一个Collection，类似于一个对象，有很多属性，而这些属性值又可以是另一个对象。本系统各个Collection抽象图如下：

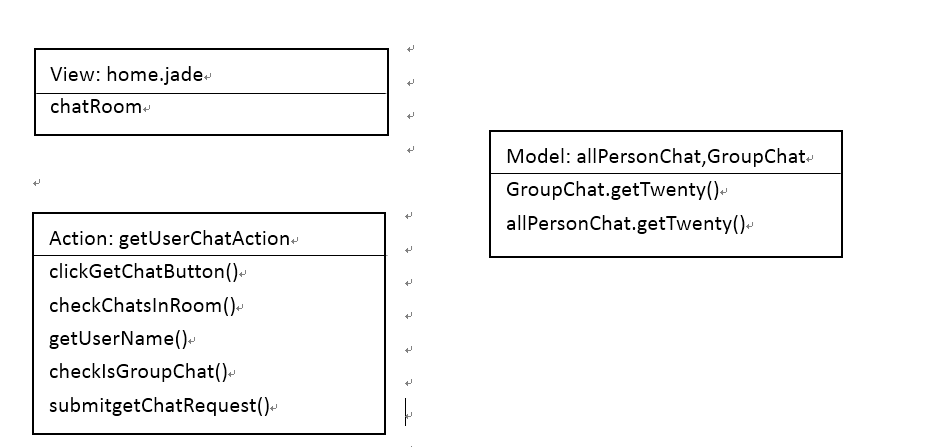


## 4.3 用例分析

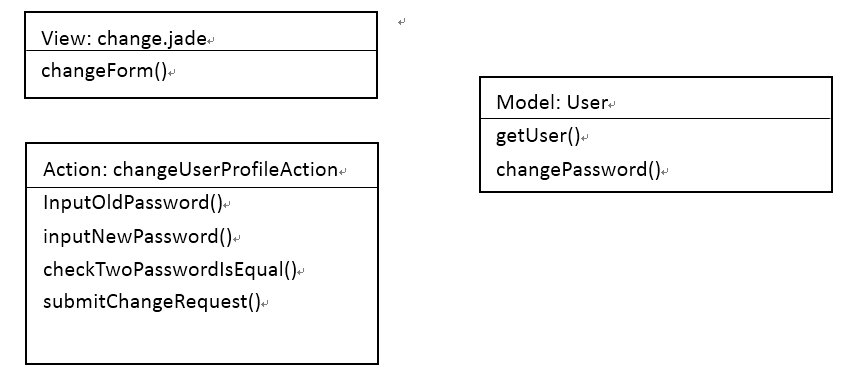
### 4.3.1 登录用例的类分析和设计



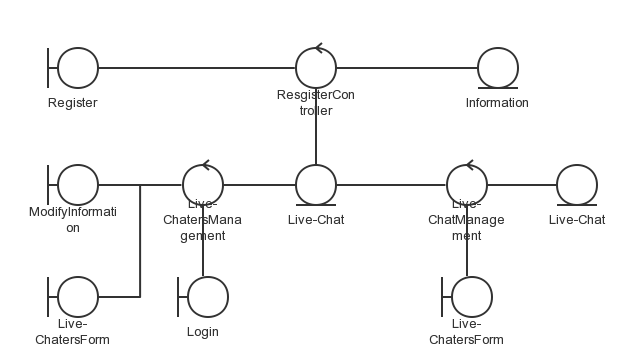
### 4.3.2 获取用户聊天记录用例的类分析和设计



### 4.3.3 修改个人信息用例的类分析和设计

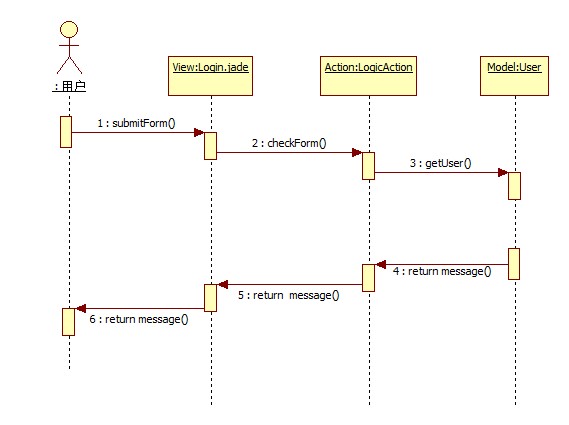


## 4.4 总类图



## 4.5 Live-Chat 用例顺序图分析

本文用例较多，选取了登录用例进行分析，可以得出下图



由登录用例顺序图可知，用户首先跟登录界面交互，输入登录信息。之后将登录信息交给逻辑动作层进行检验，逻辑动作层再将信息传给数据访问层进行查验。最后返回登录信息。

## 4.6 Live-Chat 分析类机制

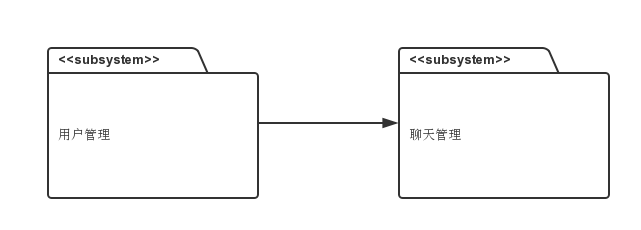
**部分分析类到分析机制的映射**

|  |  |
| --- | --- |
| **分析类** | **分析机制** |
| login.jade | 图形化 |
| logicAction | 资源管理 |
| User | 持久化，安全性 |
| Home.jade | 图形化 |
| getUserChatAction | 资源管理 |
| allPersonChat | 持久化，安全性 |
| GroupChat | 持久化，安全性 |
| Change.jade | 图形化 |
| changeUserProfileAction | 资源管理 |

## 4.7 子系统设计

### 4.7.1 子系统划分

本节基于4.1系统框架和4.2系统关键抽象的基础，将系统划分成两个逻辑上相对独立子系统，分别为：用户管理子系统，聊天管理子系统。每个子系统包括了表现层、业务层和数据层。下图显示了两个子系统间的相互关系：



### 4.7.2 用户管理子系统

本节以用户管理子系统为例，介绍子系统设计。

用户管理子系统实现了用户实体的业务逻辑功能，如：用户登录，用户注册和用户注销等功能。

依据4.1系统架构以及用户实体的业务逻辑功能，可将用户管理子系统分成三个子模块，包括：用户表单，用户管理以及用户实体，分别对应于4.1系统框架的表现层，业务层以及数据层。

1) 用户表单子模块

对应于4.1系统框架的表现层，该模块主要包含了用户表示层的各种表单，如：用户登录表单，用户注册表单和用户注销表单等。

2) 用户管理子模块

对应于4.1系统框架的控制层，该模块主要关于用户实体的各种业务逻辑类，如：查找用户，添加好友等功能。

3) 用户实体

对应于4.1系统框架的实体层，该模块包含了用户实体类(User.java)，实现数据的持久化。

### 4.7.3 分析类到设计元素映射

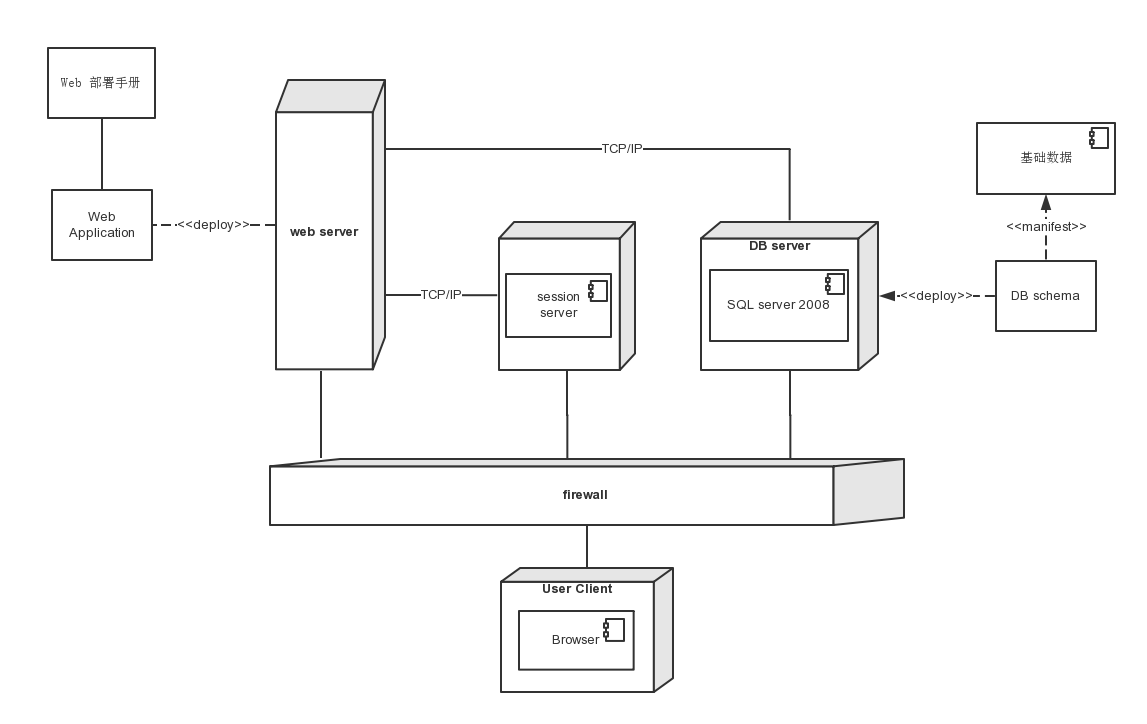
|  |  |
| --- | --- |
| 分析类 | 设计元素 |
| 用户实体功能 | 用户管理子系统 |
| 聊天实体功能 | 聊天管理系统 |

### 4.7.4 设计元素及其包

由于系统的五个子系统逻辑上相对独立，所以将每个子系统设置为一个包。

|  |  |
| --- | --- |
| 设计元素 | 包 |
| 用户管理子系统 | 用户管理包 |
| 聊天管理子系统 | 聊天管理包 |

## 4.8 部署图



## 4.9 构件图

