# 详细设计

1. 概述
   1. 系统简述

本即时通讯系统运用共享数据结构技术及多线程技术，通过I/O端口完成数据转发功能，在不同客户端间实现基本的数据通信功能。是一个方便用户聊天、通讯的桌面应用程序

* 1. 软件设计目标

本软件的设计目标是最终实现一个具有良好用户交互体验和基本通信功能的交流软件。具体包括：用户注册与管理功能、个人信息管理功能、好友系统与实时聊天功能（仅图片）。本软件主要用于研究性学习而非实用，对安全性的要求不严格。在性能方面，应保证正常实用状态下无明显卡顿，并将信息传递与展示的时延控制在合理范围内

* 1. 参考资料

《Applying UML and Patterns》

《Software Architecture in Practice》

* 1. 修订版本记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本修订历史记录 | | | | | |
| 序号 | 修订内容 | 修订人 | 修订日期 | 版本号 | 备注 |
| 1 | 初版编纂 |  | 2018-10-31 | 0.5 |  |
| 2 | 增添成稿 |  |  | 1.0 |  |

1. 术语表

（对本文档中所使用的各种术语进行说明。如果一些术语在需求规格说明书中已经说明过了，此处不用再重复，可以指引读者参考需求说明。）

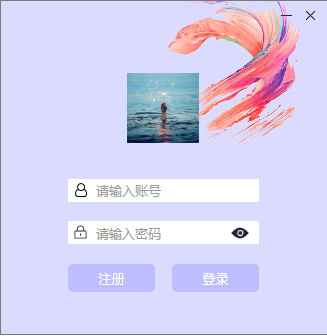
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. 设计概述
   1. 系统的复用计划

系统的复用计划：库、框架、模式、构件等方面的复用

* 1. 系统接口设计
     1. 界面设计
        1. 登录界面：

登陆界面提供账号密码输入栏，显示账户对应的头像；并设计对应按钮执行登陆功能或转入注册页面



* + - 1. 注册界面

用户可以在注册界面输入注册账号的相关信息，例如用户名、密码、需要绑定的邮箱等。用户可以手动查看输入的密码。系统在执行成功注册后返回登陆界面



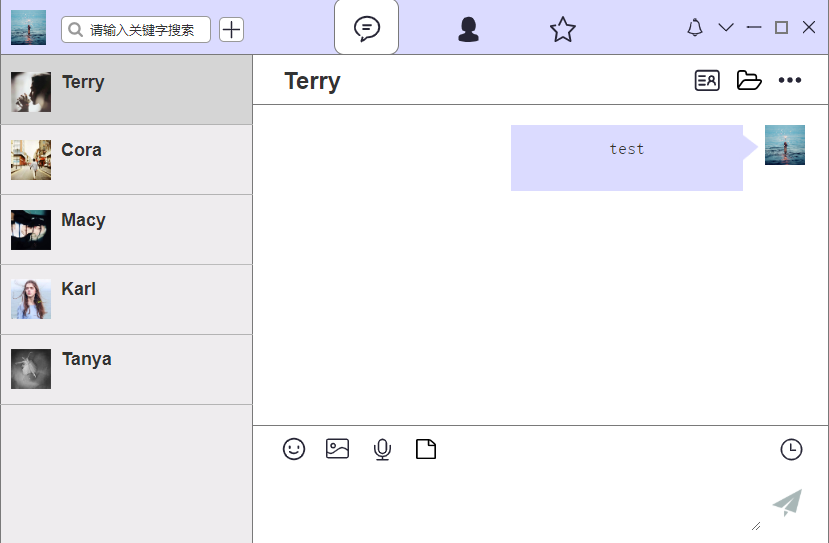
* + - 1. 主页面

主页面由顶部功能区以及下方嵌套的三个子页面组成，用户通过功能区的按钮在三个子页面间进行切换以使用所有主要功能。功能区还另外包含用户头像显示、搜索栏——意图实现统一的搜索入口。整合联系人、群、聊天记录的搜索；添加按钮——用于添加联系人或群；通知铃——整合系统提示；菜单键——转向通用设置页与注销登录。以及基本的页面控制控件



1. 聊天框子页面

聊天框子页面的左侧固定显示会话列表，用户可通过点击某一个会话在右侧打开对应的通信页面，执行基础的文字收发功能。其它设计中的功能包括表情收发、图片收发、语音收发、文件收发等



1. 联系人子页面

联系人子页面左侧固定显示联系人列表，按照姓名首字母排序。用户可通过点击某一个联系人在右侧查看对应的信息，且可直接转入通信界面



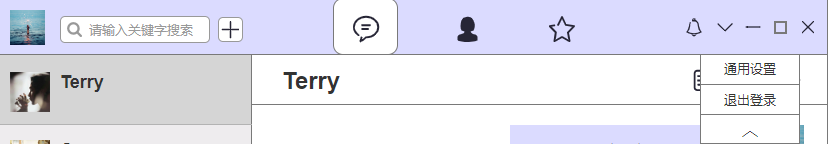
1. 工具箱子页面

工具箱子页面整合个人资料、我的群聊等功能。用户可以在个人资料页查看昵称、地区与账号等信息，并可对个人资料进行编辑。其它设计中的功能包括收藏夹、我的文件等



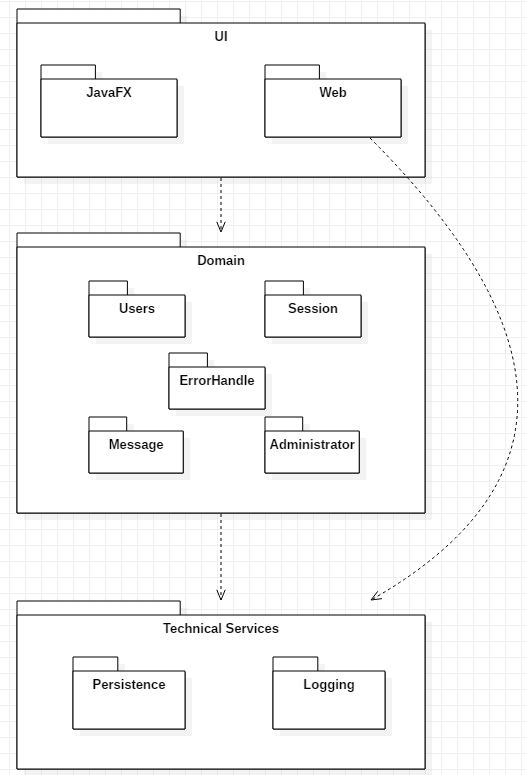
* + - 1. 通用设置页面

用户可以在菜单页面中选择进入通用设置页。并在其中调整一些交互特性、产品功能已经提供反馈等。此页面尚在设计中



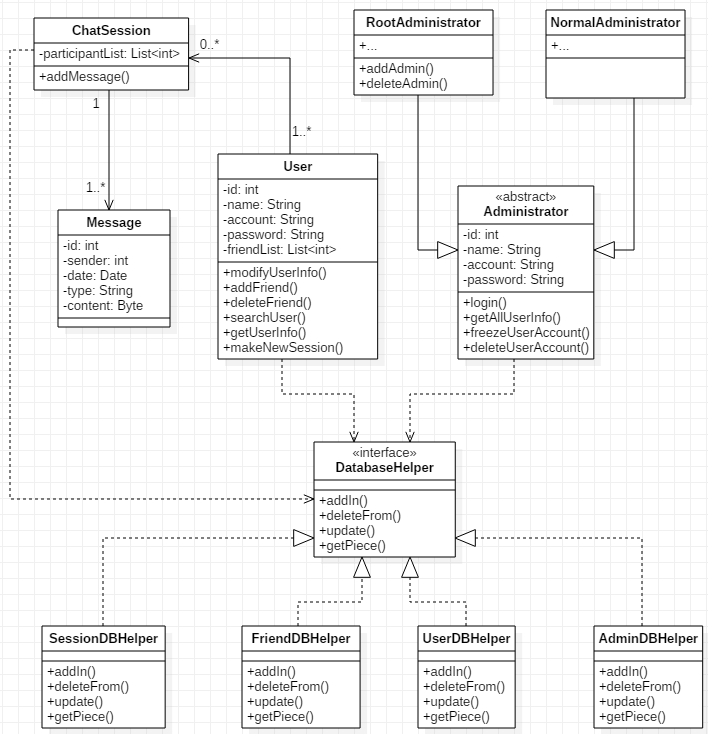


* + 1. 系统外部接口设计：与外部系统的交互设计
    2. 系统内部接口设计：各子系统、各模块间的接口设计
  1. 对象模型设计

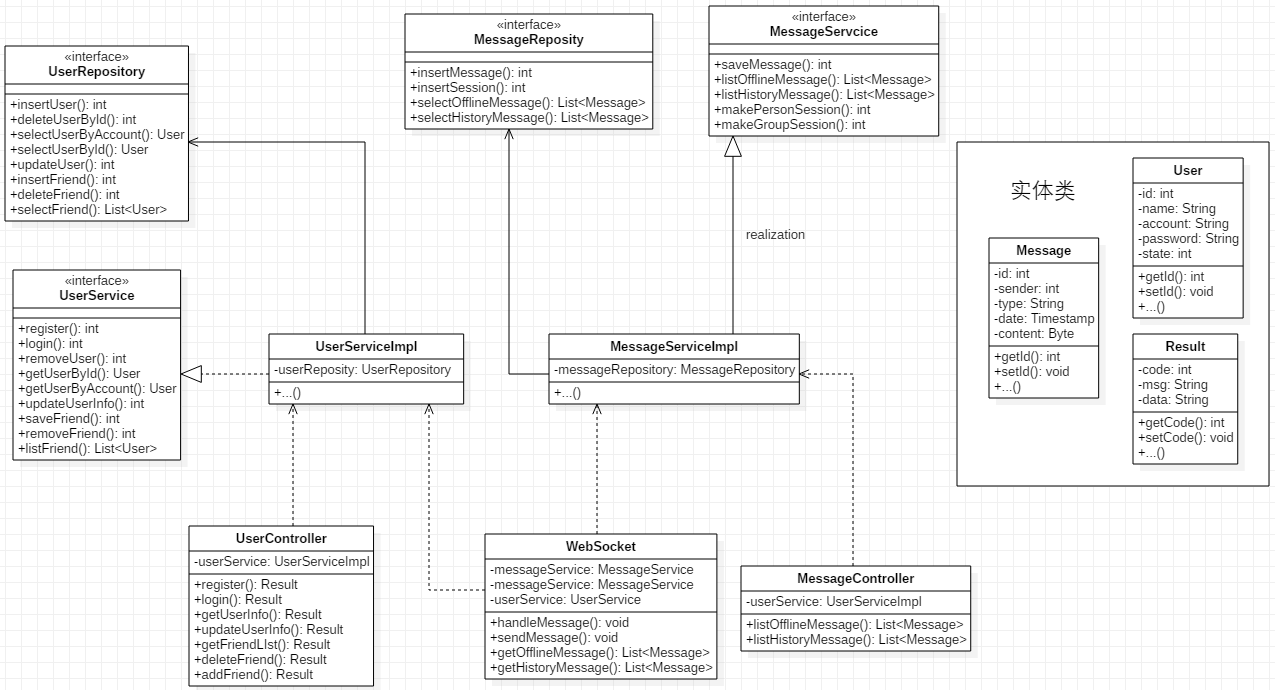


系统层级设计如左图，主要分为3层：UI层、领域层与技术支持层。

1. UI层主要通过以WebSocket实现的Web通信功能完成客户端与服务器的交流。并通过基于JavaFX实现的UI类合理地将信息呈现给用户
2. 领域层揭示了本系统的功能核心：处理几个平行模块：用户、会话、管理员与消息间的相互关系。并实现一定程度的错误处理功能
3. 技术支持层主要包含信息的持久保持与为了实现可调试可恢复而必须的系统日志的生成，这一部分的实现主要依托于所使用的数据库技术与服务器端技术



系统对象模型如上图，包含User、Message、ChatSession等对象，对应用户、消息、会话等实体。由于本系统涵盖客户端与服务端，单一System实例无法良好的表示需求。故此模型实际上是一种概念上的抽象，表达了对象实体应具有的属性以及不同对象间最终期望达到的一些交互。例如增删查改、对应的数据库操作



以最基本的用户与消息为例。上图中一个User类在实现上分为4个部分。首先是用于储存信息的实体类，用于在内存中储存提取到的数据，其属性字段与底层数据库最相似。随后自底向上分别是UserRepository：这个类中的函数用于对数据库进行操作；UserService & UserServiceImpl：接口实现类中的函数处理基础的事务逻辑，实现诸如移除好友、登陆一类的功能；UserController：这个类存在的最主要目标是增加一层间接性，利于逻辑分离，它的函数功能最接近用户请求，可能是Service中事务逻辑的组合

* 1. 系统用例实现详细设计

1. 顺序图
2. 状态图
   1. 系统非功能设计