软工大作业 - 即时通讯系统 需求文档

> Last Update @ 2024-02-27

项目基本情况

IM (Instant Message)，即在线聊天，是一种高效的沟通方式，支持两个或多个参与者在网络上通信。

请设计一个即时通讯系统，提供类似 iMessage 的核心聊天功能，并保证关键的实时性、数据完整性。

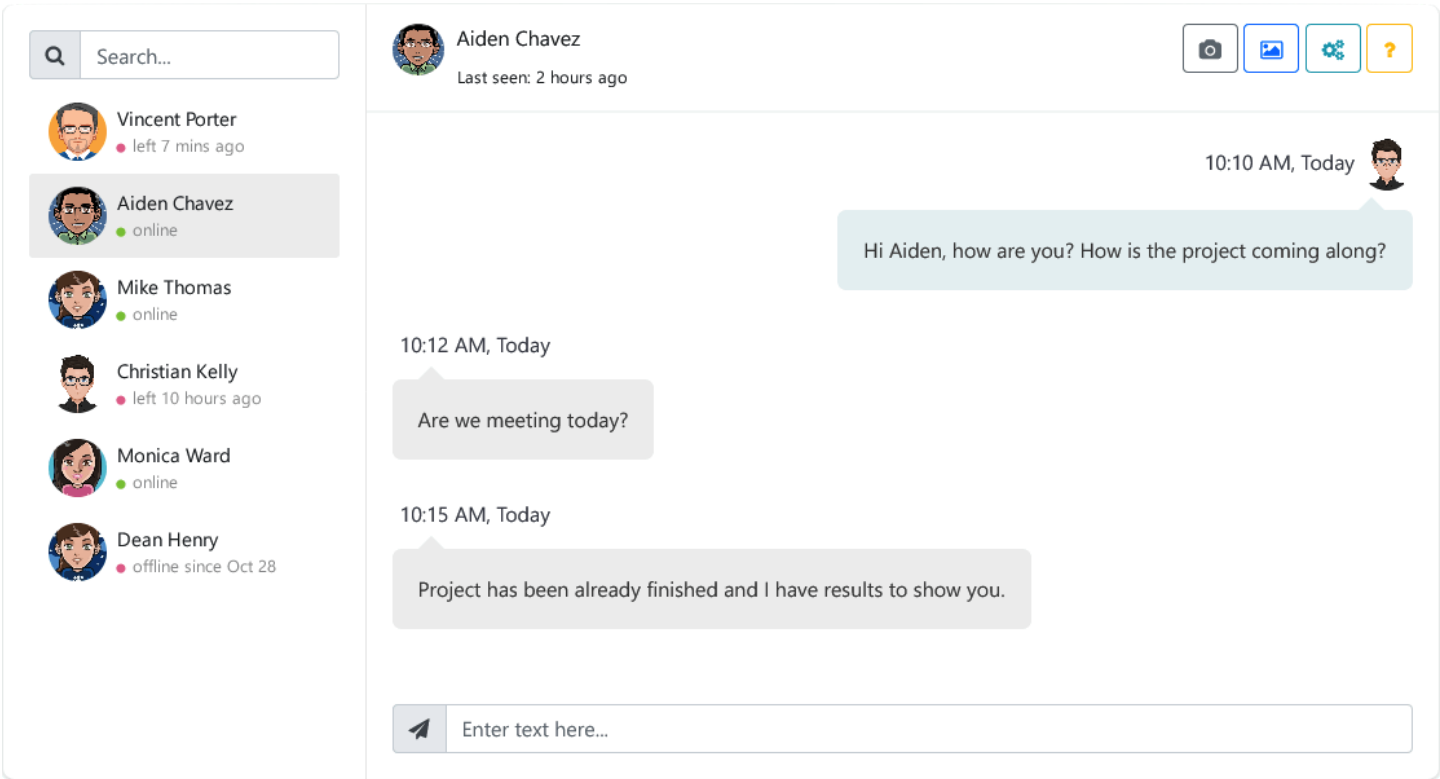
基本功能说明

本项目要求的功能包括：

- 用户管理功能。需要支持用户注册、登录功能，以及实现好友系统等
- 在线会话功能。需要支持用户间私聊以及用户群聊功能，同时需要满足信息安全等即时通讯系统的基本要求

具体功能的要求见后续功能评分列表。

本项目要求支持 PC Web 端，基本的 UI 框架类似下图。其中左侧为会话列表，需要列出当前可用的会话，右侧为会话界面，具体展示会话中的消息：



项目评分细则

本项目评分框架如下所示：

- （40%）系统基本要求
  - 本项目的基本要求包括**可用性**、**安全性**、**实时性**和**数据完整性**
  - 该部分评分具体见后文描述，在验收时仅需要系统表现符合功能描述即可获得功能点全部分数
- （50%）功能评分
  - 该部分评分具体见后文描述，在验收时仅需要系统表现符合功能描述即可获得功能点全部分数
- （10%）开发文档
  - 要求在项目开发完毕后提交一份项目说明文档
  - 最终呈现的说明文档至少应当包含下述内容
    - 需求分析。要求罗列用户故事，绘制适当的图表（如用例图、泳道图等）说明项目工作流程
    - 模块与 API 文档。写明项目开发中设计的各个模块及其用途，写明前后端之间所有 API 的文档和交互格式，必要时使用如流程图、序列图、状态机等图表说明。要求达到后续维护与开发人员可以直接通过该文档了解该项目设计和开发基本状况的程度
    - 数据库设计文档。要求列出数据库设计中所有的表和表间关系
- （额外扣分项）过程管理
  - 在实际开发过程中，如有违反开发规范的情况将会酌情扣分。细则见网络学堂公布的《大作业过程分细则》

系统基本要求

本项目基本要求列表如下，所有要求赋分总和为 40 分。

需求	功能描述	赋分
可用性	最终提交的网站能够正常显示	10
	前后端能正常通信	10
	SEcoder 平台上 CI/CD 工作正常	10
安全性	严格要求数据库敏感信息以加密方式存储	2
	禁止在不安全信道上明文传递敏感信息	2
	应当在所有需要判定请求发起者身份的接口处采用合理的鉴权手段维护数据安全	2
实时性	要求消息送达的平均延时在 200 ms 以内	2
数据完整性	要求设计并实现机制保证消息可靠、完整送达	2

功能评分

本项目功能评分列表如下，所有功能赋分总和为 50 分。

首先是用户管理部分的功能表，共计 17 分：

一级需求	二级需求	功能描述	赋分
基本管理	用户注册	要求能够令用户注册，需要对用户填写的注册信息提出合理的格式要求	2
	用户注销	已登录的用户可以选择注销在该系统中自己的账号，注销后服务器端应当正确处理涉及到该用户的所有数据，如聊天记录与好友关系等	1
用户认证	登录登出	用户能够使用已经注册的账号和正确的认证信息登录系统，并可以登出系统。用户登录后应当和服务器端同步消息记录	3
	信息编辑	已登录的用户可以修改自己的用户名、密码、头像、邮箱、手机等个人信息。对于涉及到身份认证信息的修改（如密码、邮箱、手机等），应当至少通过一种认证手段验证身份	2
好友关系	用户查找	允许用户通过用户名等关键词查找其他用户并浏览其基本信息	2
	好友申请	至少允许用户通过用户查找、群聊两种方式申请添加另一用户为好友，经对方同意后建立双向好友关系	2
	好友删除	允许用户删除好友	2
	好友列表	需要展现用户的好友列表，允许用户对自己的好友分组	3

之后是在线会话通用部分的功能表，共计 19 分：

一级需求	二级需求	功能描述	赋分
基本要求	界面显示	聊天界面至少需要展示参与人的头像、所发送的信息，需要对用户本人发出的信息在显示上特殊处理，如消息气泡颜色不同	1
	发送信息	允许用户在聊天界面发送文本信息	4
	辅助信息	对每一条信息合理展示辅助信息，如发送时间、对方是否已读（针对私聊）、已读该消息的成员列表（针对群聊）等	1
	未读计数	每一个会话，包括私聊和群聊，均需要展示该会话目前未读消息的数量	2
拓展列表	基本要求	右键或鼠标悬浮某条消息会弹出拓展列表	1

	回复消息	拓展列表中包括回复功能，允许用户回复某条消息（2 分）。被回复的消息应当展示该消息被多少条消息所回复，单击回复某条消息的消息应当可以跳转到其所回复的消息处（1 分）	3
聊天记录	查看记录	允许用户查看该会话下所有的聊天记录	4
	筛选记录	允许用户通过时间/发送成员（针对群聊）等标准筛选聊天记录	2
	删除记录	允许用户删除聊天记录	1

然后是在线会话群聊部分的功能表，共计 14 分：

一级需求	二级需求	功能描述	赋分
基本要求	列表创建	允许用户从好友列表中选取部分好友创建群聊	2
	信息展示	应当设置某一个界面展示当前群的基本信息，至少需要包括群聊名称、群成员、历史群公告等	4
群聊管理	群管理员	创建群聊的用户默认为群主，群主可以在群成员中指定管理员	2
	群主转让	群主可以主动将群主身份转让给其他群成员	1
	移除群员	群主与群管理员有权限直接移除指定群成员，群主可以移除自己以外的任何群员，群管理员仅能移除非群主、非群管理员的群成员	2
成员管理	成员邀请	群成员可以邀请自己的好友加入群聊，需要经过群主与群管理员的审核	2
	成员退出	群成员可以自由退出群聊	1

预备知识

长连接

- 客户端和服务端之间操作可能比较频繁，保持一个长连接有助于节省资源
- 服务端在主动给客户端推送消息时，可以使用保持好的长连接，有助于提高消息送达及时性

消息同步

- 向在线用户推送消息
- 为离线用户保存“脱机”消息，用户登录时可以从服务端获取“脱机”消息
- 可以学习参考**读扩散**和**写扩散**这两个概念

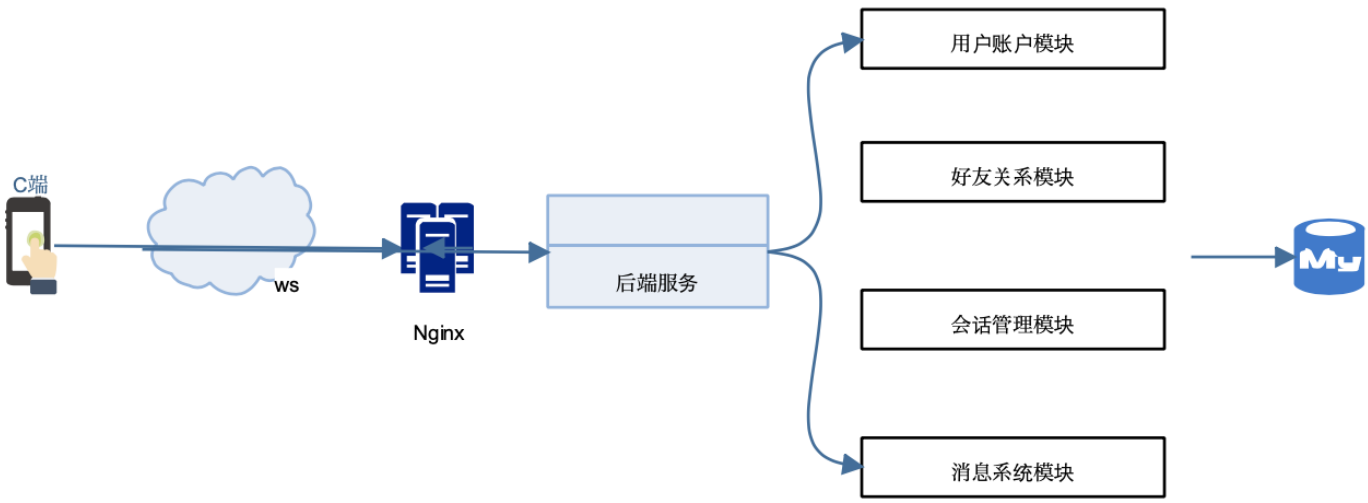
其他参考资料

- 客户端和服务端的消息同步是个难点
  - https://cloud.tencent.com/developer/article/1194222
- 保证消息的可靠送达是个难点
  - http://www.52im.net/thread-294-1-1.html
  - http://www.52im.net/thread-594-1-1.html
- WebSocket

简单设计

总体架构图

我们推荐的一种可能可行的架构设计为：



服务器存储选型

这里以 MySQL 为例，将为了完成核心功能所需要建立的数据库表给出参考，在实际实现该项目的时候所实际建立的数据库架构可能与下述参考相差较大：

用户账户

- 全局唯一的用户标志符 `user_id`，整型数据
- 用户名 `name`，字符串数据
- 密码（加密后） `password`，字符串数据
- 注册时间 `register_time`，时间戳
- 登陆时间 `login_time`，时间戳

好友关系

好友关系是双向的，A 申请添加 B 为好友，B 同意后，A、B 才会出现在双方的好友列表中，否则均不出现。

- 全局唯一的用户标志符 `user_id`，整型数据
- 好友的用户标志符 `friend_user_id`，整型数据
- 更新时间 `update_time`，时间戳

会话管理

首先是保存保存会话基本信息的会话表：

- 全局唯一的会话标志符 `conversation_id`，整型数据
- 会话名称 `conversation_name`，字符串数据
- 创建时间 `create_time`，时间戳
- 更新时间 `update_time`，时间戳

之后是保存会话的成员的会话成员列表，及每个成员在会话中的配置信息：

- 全局唯一的会话标志符 `conversation_id`，整型数据
- 成员的用户标志符 `member_user_id`，整型数据
- 成员的加入时间 `join_time`，时间戳
- 更新时间 `update_time`，时间戳
- 成员当前已读到的消息 index `read_index`，整型数据

消息系统

该系统是这个项目的核心点，需要做好服务端消息的存储，客户端与服务端消息的同步。详情可参考先前给出的参考资料。

服务端消息存储

首先需要保存每个消息的内容：

- 全局唯一的消息标志符 `msg_id`，整型数据
- 消息内容 `msg_body`，字符串数据

之后需要保存每个会话中，收到的所有消息列表：

- 全局唯一的会话标志符 `conversation_id`，整型数据
- 全局唯一的消息标志符 `msg_id`，整型数据
- 消息创建时间 `create_time`，时间戳
- 消息发送方用户标志符 `sender_id`，整型数据

客户端与服务端的消息同步

这里需要管理、保存每个用户的收件箱，可以收到来自多个会话的消息：

- 全局唯一的用户标志符 `user_id`，整型数据
- 全局唯一的会话标志符 `conversation_id`，整型数据
- 全局唯一的消息标志符 `msg_id`，整型数据
- 更新时间 `update_time`，时间戳

同步时机可能包括以下三处：

- 客户端本地无数据，第一次从服务端获取数据
- 客户端本地有数据，从服务端获取离线增量消息
- 客户端在线，服务端主动给客户端推送增量消息

简单示例

为了帮助大家快速上手 IM 系统，这里我们提供一份来自字节跳动的 IM 系统实现参考文档。该文档是 PDF 格式，作为本大作业说明文档的辅助文档下发。

附录

用户体验

作为一个完整的软件工程项目，在满足基本要求的基础上，我们建议大家在开发中也要适当关注用户体验。做到以下几点，能够让系统更加易于使用，也能让新用户更快地熟悉系统。

- UI 设计合理，布局合理，符合 UI 设计惯例
- 交互性强，应用行为符合常理，对用户误操作有友好提示
- 网页能够及时响应用户操作，尽可能避免操作卡顿、用户等待事件的发生

IM 系统的高级功能

在本项目的功能表之外，如果你有足够的兴趣和时间探索项目的更多可能性，可以自由地进行尝试和实践。

以下列出的是一些常见于主流 IM 系统（如微信 / QQ / 飞书等）的高级功能，这些功能不属于本项目的要求，实现与否**不会影响**最终评分，仅供感兴趣的同学参考。希望拓宽技术视野的同学可以根据自己的兴趣和能力，挑战更有难度的高级功能，让你的 IM 系统变得更加完整和强大。最终实现的成果可以在大作业展示课上与大家分享。

- 多媒体消息：支持图片、语音、视频、表情、文件等信息内容
- 用户提及：用户在文本输入框内输入 `@` + 用户名后可以提及他人，被提及的用户会收到提醒
- 消息撤回：消息发出后，在一定时间内，发出者可以选择撤回该消息
- 消息转发：用户可以选择若干消息，将它们合并转发至其他会话中
- 智能对话：将大语言模型 (LLM) 能力接入 IM 系统，为用户提供定制化的智能对话服务
- 音视频通话：允许两个用户间进行实时语音/视频通话
- 移动端适配：用户可以在 Android / iOS 移动端通过 APP 访问系统

有关移动端适配，我们给出部分提示：

- iOS 端的适配较为繁琐，会面临 iOS 复杂的权限管理问题
- 一种可以参考的移动端跨平台（同时支持 Android 和 iOS）是 React Native，其语法与 React 近似