

# ChatMaker 项目报告

2400013086

2025 年 6 月

## 1 程序功能介绍

### 1.1 核心功能

- 多种消息类型支持（文字/表情/语音/图片/通话）
- 转账与红包功能
- 系统消息提示

### 1.2 消息操作

- 头像切换，提升趣味
- 文字消息可双击编辑，支持微信表情
- 消息上移/下移/删除，灵活管理

### 1.3 界面细节与导出

- 顶栏自定义名称，手机状态栏仿真
- 底部输入框支持文字与表情
- 一键导出聊天记录为图片

## 2 项目各模块与类设计细节

### 2.1 项目整体架构

项目采用**模块化设计 + 样式分离**架构，主要目录如下：

目录	描述
main.py	程序启动入口，初始化主窗口
ui/	核心 UI 模块，划分为 <b>area/</b> （复合区域组件）、 <b>widgets/</b> （基础组件）与配置文件
styles/	各 UI 控件的 Qt 样式表（.qss）
fig/	图片资源目录，细分为头像、表情、图标等子文件夹

目录	描述
config.py	全局资源配置与路径管理，采用单例模式

## 2.2 主界面与系统入口

### 2.2.1 main.py - 程序入口

- 创建 QApplication 实例
- 启动 MainWindow 并进入事件循环
- 采用 `if __name__ == "__main__":` 语法结构安全运行

### 2.2.2 main\_window.py - 主窗口

- 继承自 QMainWindow
- 使用 QHBoxLayout 管理 ChatArea（主聊天区）与 FunctionArea（功能区）
- 实现核心交互逻辑：
  - 消息插入 (`handle_insert_message(...)`)
  - 聊天记录导出为图片 (`export_chat_as_image()`)
  - 自适应窗口尺寸 (`set_adaptive_size()`)

## 2.3 界面核心区域组件 (ui/area/)

### 2.3.1 ChatArea - 聊天主区域

- 垂直布局，包含顶部状态栏、标题栏、中部消息区、底部输入与表情区
- 动画展开/隐藏表情面板
- 表情点击插入文本
- 管理所有消息的插入、展示、删除、编辑等

### 2.3.2 InputArea —输入区域

- 嵌套使用 TextInput 实现富文本输入框
- 包含发送、语音、表情等功能按钮
- 文本变动时自动切换按钮显示

### 2.3.3 EmojiArea —表情选择区域

- 加载微信风格 GIF 表情 (1-108)
- 使用 QScrollArea + QGridLayout 形成网格表情面板
- 表情点击通过信号发送插入命令

### 2.3.4 FunctionArea —功能区域

- 右侧插入功能面板
- 含头像切换、插入消息（红包/语音/图片）按钮
- ExpandButton 提供浮动功能选择面板

## 2.4 基础 UI 组件 (ui/widgets/)

### 2.4.1 输入框 TextInput

- 动态高度自动扩展，最多支持 4 行
- 支持 Enter 发送 / Shift+Enter 换行
- HTML 插入 emoji <img> 标签，实现图文混排
- 信号：
  - send\_trigger: 发送消息
  - text\_input: 控制发送按钮显隐

### 2.4.2 聊天气泡 BubbleWidget

- 支持双击编辑、光标位置插入 emoji
- 富文本支持 + 自动尺寸调整
- 类型拓展：
  - VoiceBubbleWidget: 语音/通话消息
  - ImageAndGifWidget: 图片/GIF 消息
  - Pocket: 红包与转账模拟消息

### 2.4.3 消息单元组件

- MessageWidget: 聊天消息展示容器（含头像、气泡、操作按钮）
- SystemMessageWidget: 时间/红包领取等系统提示
- 支持气泡拖动上下、删除、拒收等操作
- 悬浮操作按钮组实现交互管理

### 2.4.4 功能按钮组件

- ExpandButton: 鼠标悬浮弹出功能面板
- InsertButton: 图文插入按钮，自定义样式与交互动画

### 2.4.5 头像系统

- AvatarWidget: 圆角头像，响应配置更新
- DoubleClickAvatarWidget: 支持双击更换头像
- AvatarSelectorWidget: 用户/对方身份切换面板（头像 + 单选按钮）

### 2.4.6 状态栏与标题栏

- StatusWidget: 模拟华为/苹果手机状态栏，支持图标配置、模式切换、图标组动态刷新
- TitleWidget: 聊天顶部标题栏，内嵌可编辑标签 LineEditWidget

## 2.5 资源与配置管理

### 2.5.1 ConfigManager —全局资源单例

- 管理头像、图标、表情、红包、状态栏图标等路径
- 提供统一 getter 接口
- 使用信号机制通知界面资源变更
- 保证路径有效性，避免图标丢失

## 2.6 样式系统 (styles/)

- 使用 .qss 文件统一管理 UI 控件样式
- 支持 hover、press、focus 等状态样式
- 各模块通过 load\_style() 动态加载对应样式文件
- 保持样式与逻辑分离，增强项目美观性与可维护性

## 2.7 模块间信号与交互机制

模块	发射信号	接收者	说明
InputArea	send_clicked	ChatArea	触发文本消息发送
EmojiArea	emoji_selected	InputArea	插入表情
FunctionArea	insert_clicked	MainWindow	插入功能性消息
ConfigManager	avatar_changed	AvatarWidget 等	动态更新头像
MessageWidget	text_deleted 等	ScrollArea	消息管理操作

## 3 小组成员分工情况

王梓 2400013086

消息文字 + 表情换行修复

状态栏设计

消息三角形效果初稿

红包转账消息实现

消息拒收实现

UI 细节调整

## 郑佳欣 2400013136

语音消息/通话/视频通话实现

实现系统消息类型和编辑

聊天截图导出功能

## 程一哲 2200011426

程序框架搭建

消息气泡上下移动及删除

用户自定义头像及消息切换

功能栏设计

聊天顶栏设计

微信表情插入

图片、表情消息实现

## 4 项目总结与反思

- 开发过程中的主要问题与解决方案
  - 初期聊天气泡存在文本溢出与换行错位，后通过字体度量计算和布局策略优化，成功实现内容自适应气泡尺寸。
  - 表情插入后难以与文本对齐，换行时无法精准度量表情大小导致换行错乱，使用 HTML 富文本标签并设置标准尺寸与对齐方式，保证表情显示一致。
  - 项目前期设计不够充分，接口划分与功能边界不明确，导致部分功能有开发重复或冲突的风险，引入 Git 分支管理规范与持续集成策略，避免多人协作冲突。
- 个人收获与提升
  - 通过本项目，团队成员系统学习了 PyQt5 的 GUI 开发流程，提升了代码规范意识和多人协作能力。更重要的是，我们深刻体会到项目设计与结构规划在开发初期的重要性，为今后的软件开发实践打下了坚实的基础。
- 继续优化想法

- 增加错误处理与异常提示机制，提升系统健壮性。
- 增加群聊系统，丰富功能种类