



Web 前端技术基础课程设计

题 目： 基于 Flutter 前端框架的

消息聚合与即时通信系统网站设计

基于 Flutter 前端框架的 消息聚合与即时通信系统网站设计

1 前言

1.1 Flutter 框架介绍与展望

1.1.1 什么是 Flutter

Flutter 是由 Google 开发的 UI 工具包，用于通过单个代码库为移动，Web 和桌面设计精美的，本机编译的应用程序。Flutter 可与现有代码配合使用，并且免费与开源。

1.1.2 Flutter 框架优点

1、Flutter 具有跨平台特性，目前已正式支持 IOS、Android、Web 三端，且对于 Windows、Linux、MacOS 等桌面操作系统提供实验性支持。

2、支持局部热重载，可提高开发效率。

3、采用自绘 UI 引擎，在不同平台上做到高性能运行。

1.1.3 Flutter 框架应用情况

Flutter 目前已经在 Google 内部广泛用于 Assistant、Stadia、Cloud Search 以及 Blogger 等项目。在 Google 之外，字节跳动、Grab、Nubank 以及 MGM Resorts 等公司也都已采用 Flutter，并通过 Flutter 提升了生产力和灵活性。其中，国内的大量互联网公司也已经开始采用 Flutter 作为开发工具。淘宝闲鱼技术团队与字节跳动团队对 Flutter 进行了大量的技术实践。

1.1.4 Flutter 框架应用前景

Flutter 作为一个跨平台 UI 开发工具包，目前已经具有一定的实用性，但与 Web 传统开发技术相比仍有所不足。对比 Vue 与 Nodejs 等 Web 开发框架，Flutter 的第三方应用库较少，目前仍在快速增长中。从国内公司支持来看，具有广泛的应用前景。

2 总体设计

本项目采用 Flutter 框架进行前端开发，在后端使用 Nginx 作为服务器负载均衡入口，数据进行负载均衡后传输至后端 Nodejs 服务器。数据使用加密的 json 格式数据以 HTTP Get/POST 方式进行传输。对于用户身份数据，使用非关系型数据库 MongoDB 进行存储；对于通信记录等大量格式化数据，采用关系型数据库 Mariadb 进行存储。

网站主体页面首页为 <https://website/#/login>，在此页面通过 cookie 或密码登陆进行鉴权后跳转至 <https://website/#/main>，程序所有功能页面通过 Flutter 框架的局部控件树刷新技术进行刷新重绘。

3 详细设计

3.1 前端页面开发

前端页面为项目的主体页面，负责所有用户侧功能实现。

页面主体分为三部分，分为左侧状态栏，中部通知栏，右侧详细消息与编辑栏。

整体页面配色设计遵循 Material Design 设计风格，如图 3-1-1 所示。

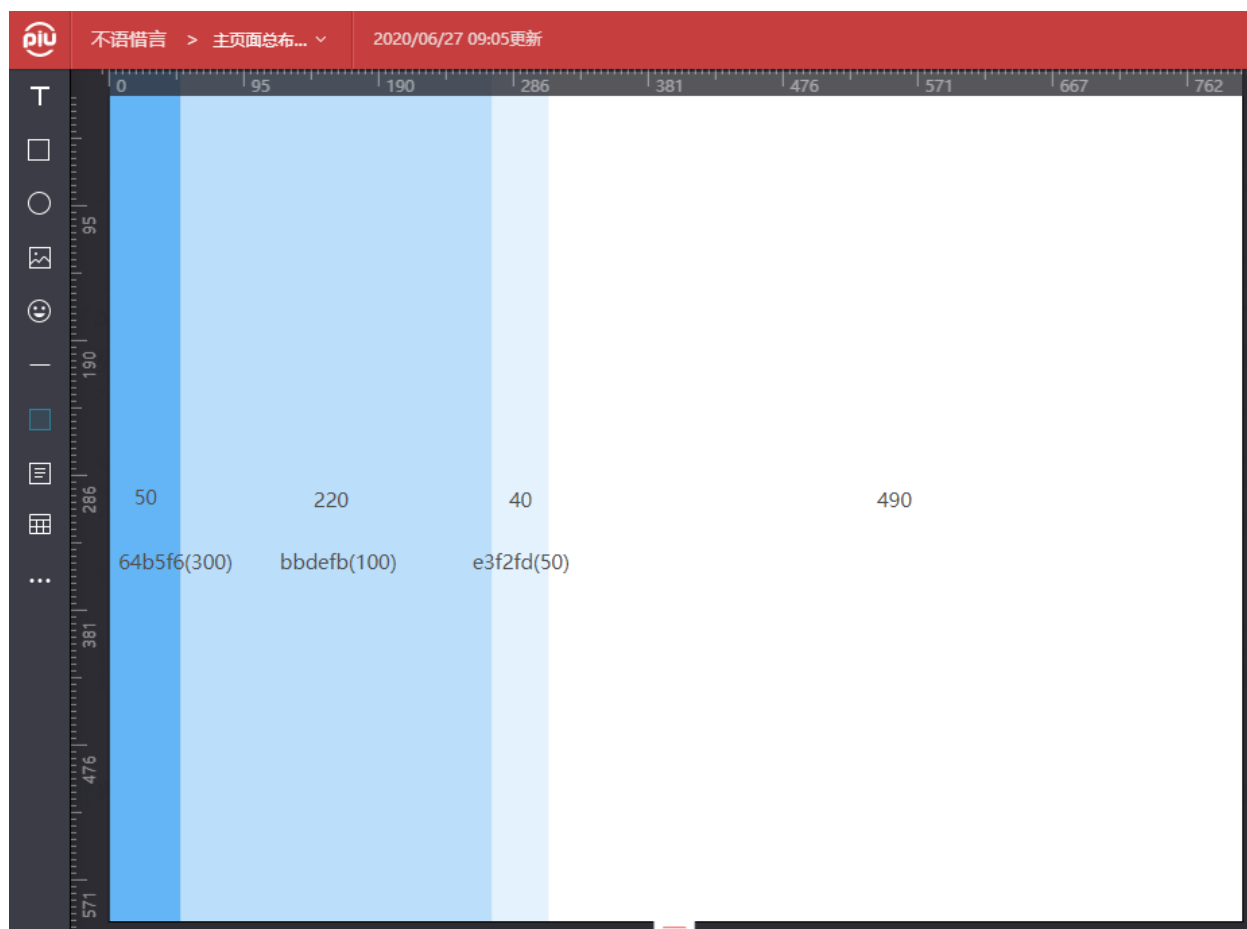


图 3-1-1 主页面总布局

左侧状态栏分为三个区块，分别为用户头像按钮，消息处理按钮，文件检索按钮。用户头像按钮可跳转至用户信息页面，提供信息更改、切换用户、注销等功能；消息处理按钮可跳转至消息处理主体页面；文件检索按钮可展示所有收到的文件以及对聊天记录中的文件、语音、图片等各类数据的检索，在系统设计中支持使用 API 调用第三方检索工具进行数据的导入、导出、处理。布局如图 3-1-2 所示。

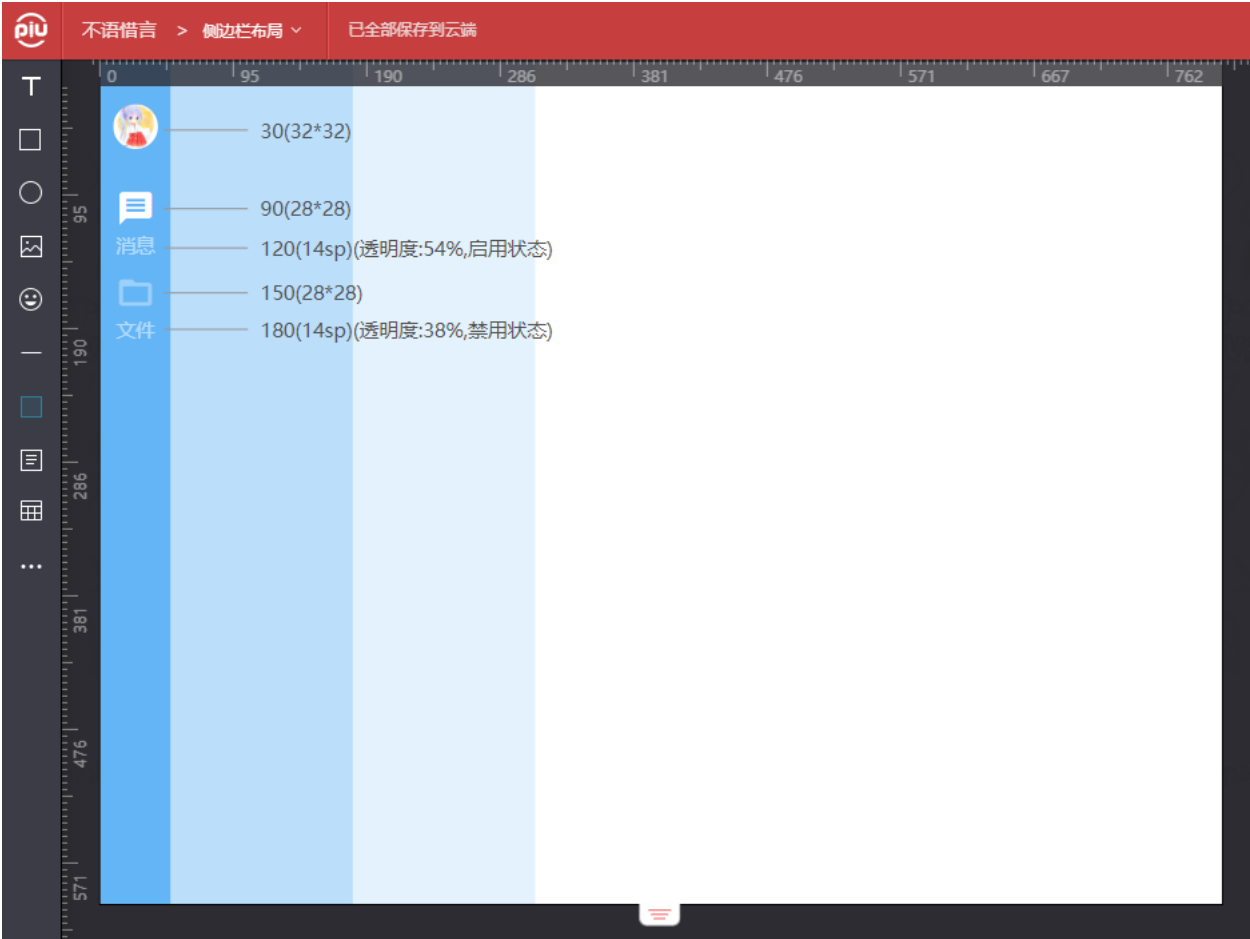


图 3-1-2 侧边栏布局

中部通知栏实现通信数据检索功能，实现了一个可进行主次切换的列表。搜索框可对数据源与联系人进行快速检索与添加。左侧列表为主要数据来源，支持使用本项目内数据处理源与导入第三方数据源，一个常见的例子是能够实现多个邮箱与多个即时通讯软件间的消息聚合。右侧列表为数据源中的单独联系人或数据项目。两个列表在选中不同焦点时会进行自动切换，便于处理不同数据源的不同信息。

右侧详细消息与编辑栏可展示每条数据项目的详细信息，对于邮箱类数据源，提供 Markdown 编辑框；对于即时通讯数据源，视软件支持情况采用不同的编辑框进行数据处理，部分数据源支持使用语音或视频通信等功能。系统本身会根据数据源支持情况显示不同功能栏按钮。

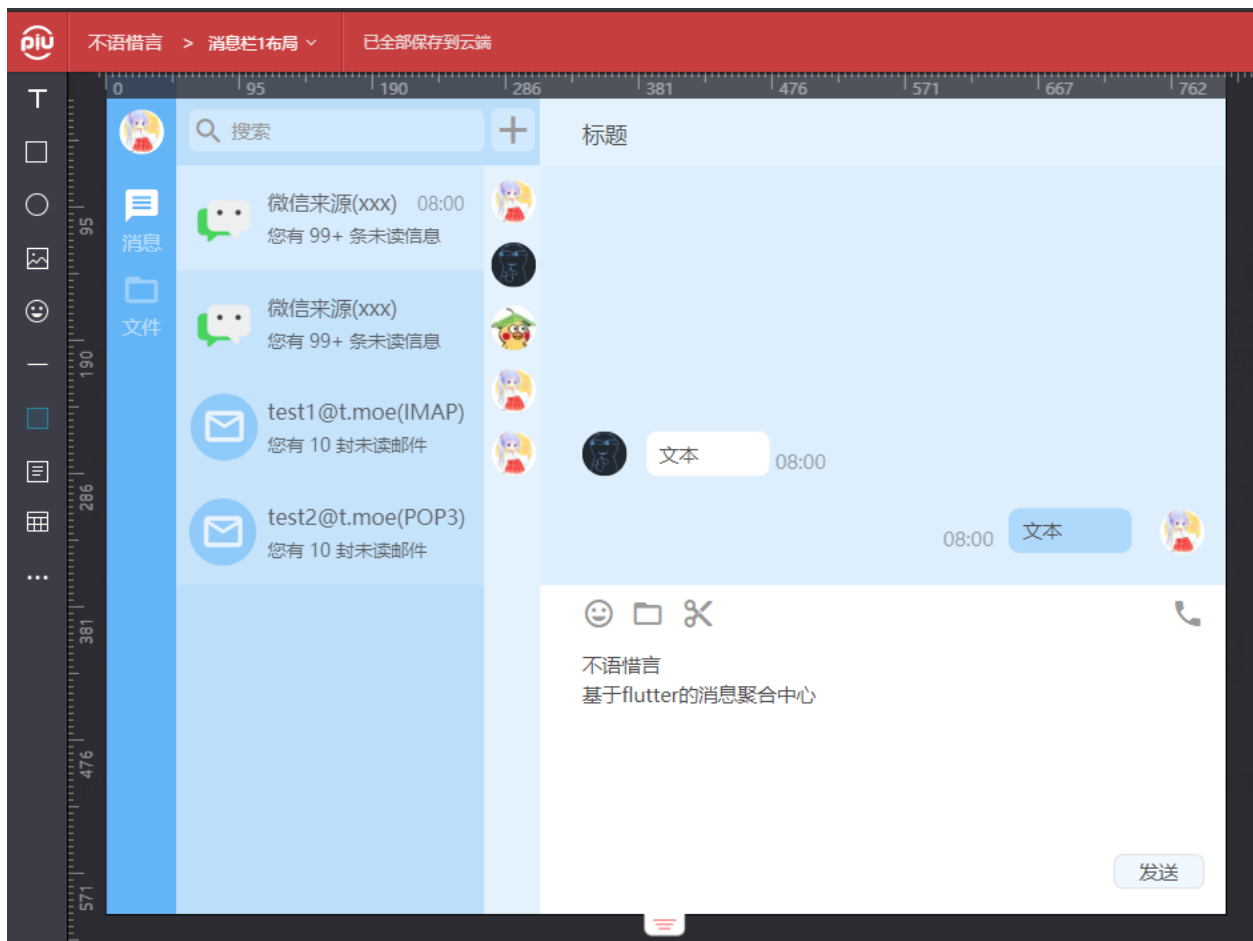


图 3-1-3 消息栏布局

由于 Flutter 框架本身的多平台特性，因此程序代码本身可以生成多平台应用程序或 Web 页面。在 Web 状态下，系统对自适应提供了初步支持，移动端竖屏未做特殊优化，显示效果较差，横屏状态可提供与电脑网页端近似的体验。下列图片为不同自适应展示。

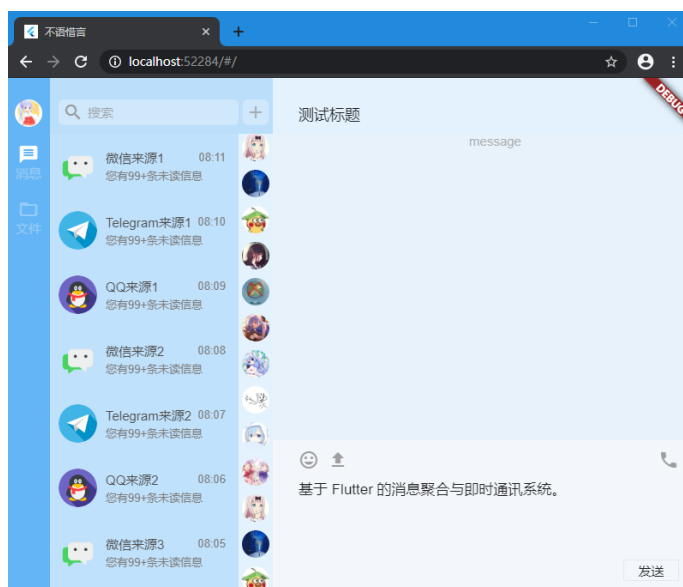


图 3-1-4 默认页面自适应状态



图 3-1-5 压缩页面自适应状态

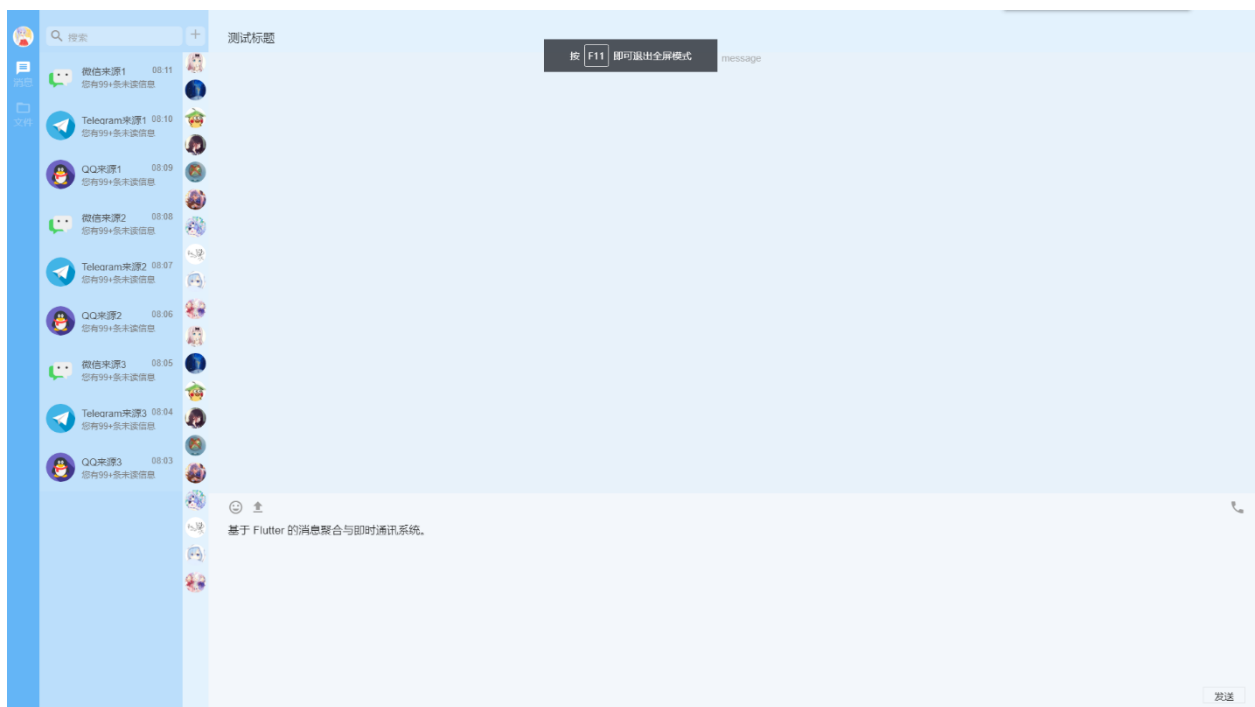


图 3-1-6 网页全屏自适应状态

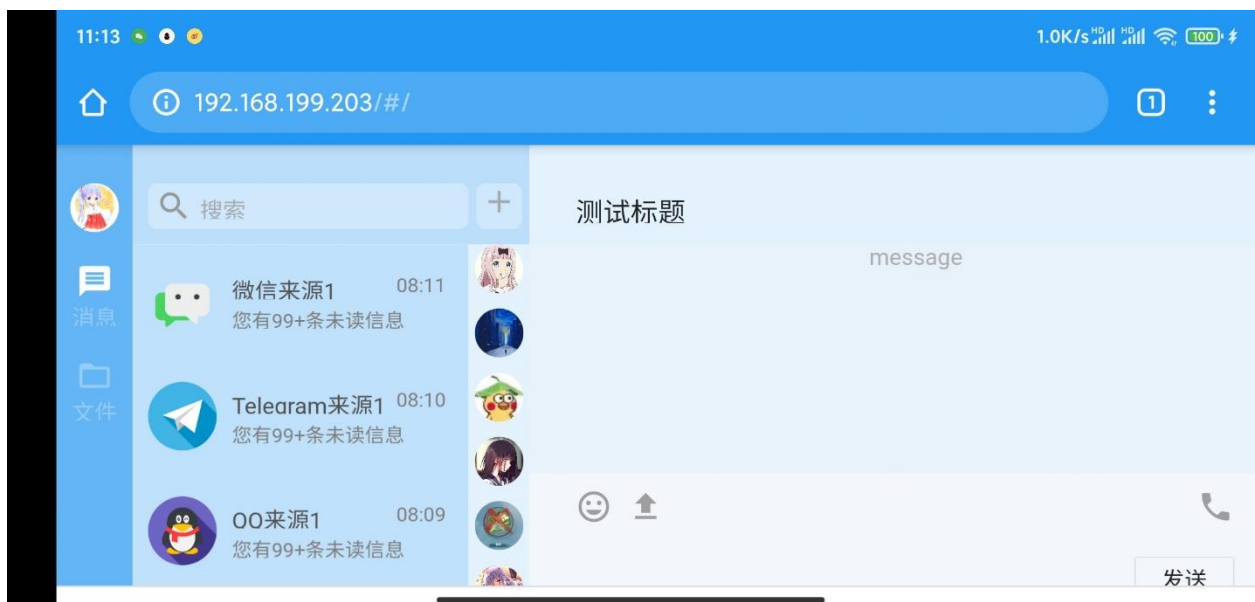


图 3-1-7 移动端横屏自适应状态

3.2 前端部分代码展示

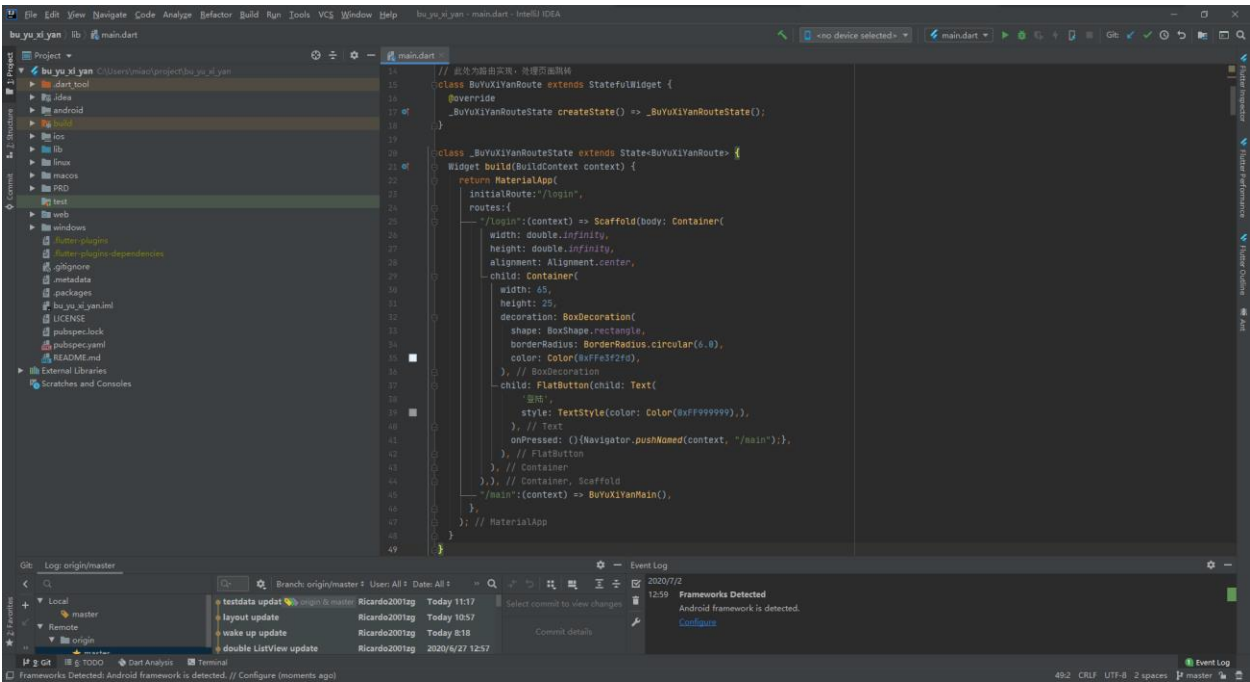


图 3-2-1 路由部分代码展示

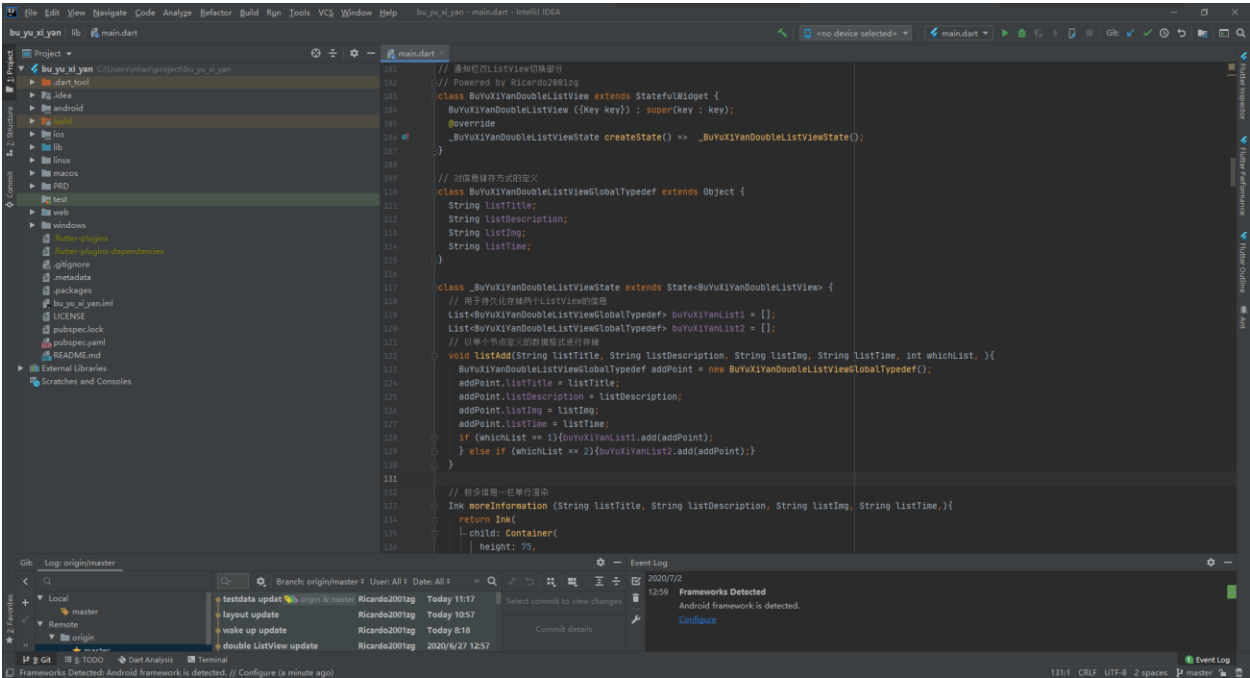


图 3-2-2 通知栏双列表切换部分代码展示

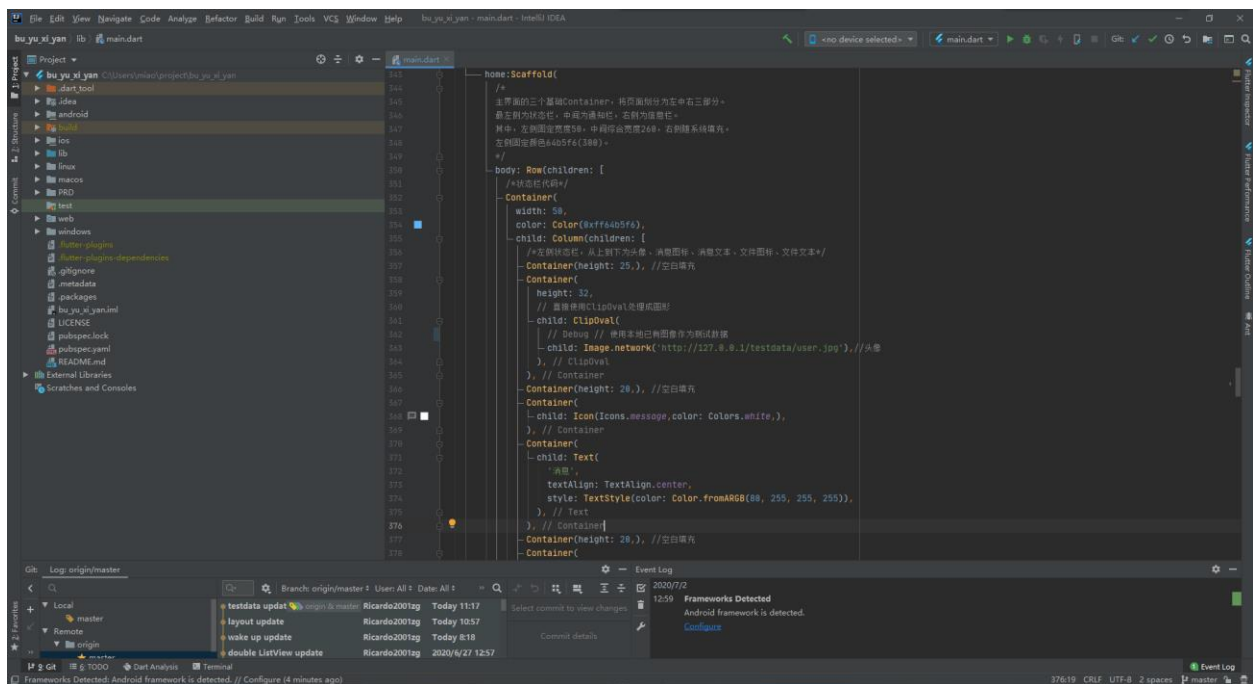


图 3-2-3 状态栏实现部分代码展示

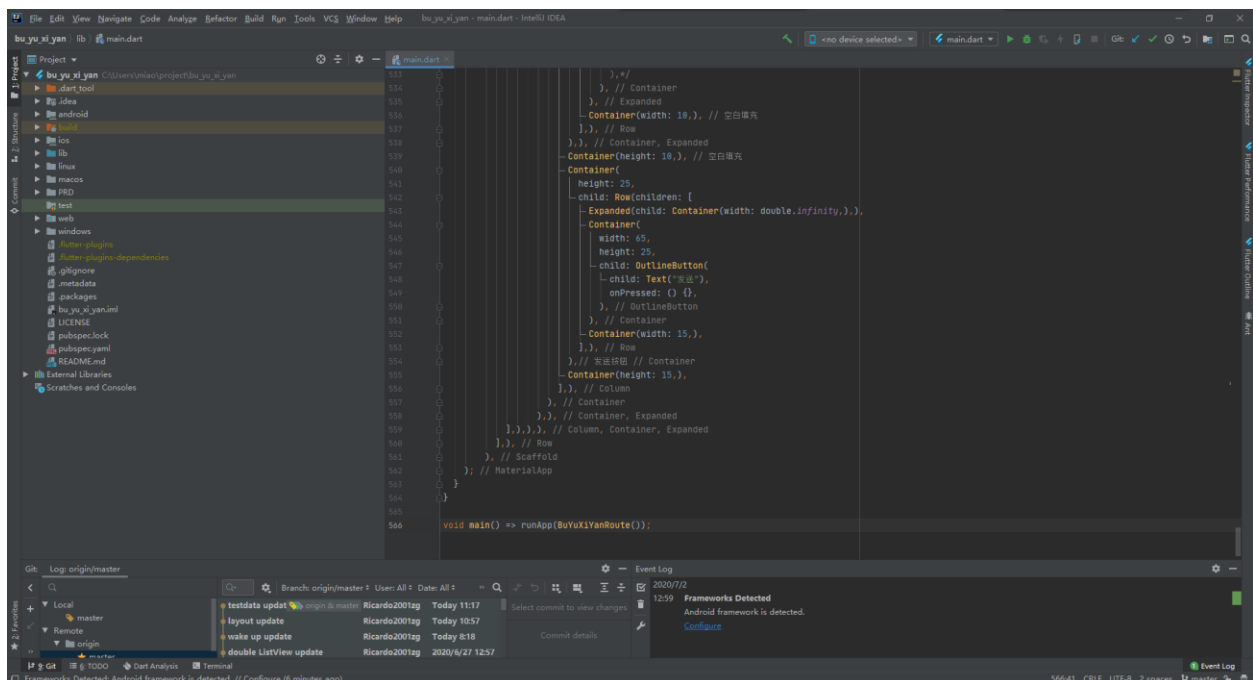


图 3-2-4 主函数部分代码展示

3.3 后端开发

目前由于不同平台的个别函数库不同，无法做到全平台适配，后端开发暂时使用加密后的 json 格式数据以 HTTP Get/POST 方式进行传输。用户系统定时对服务器系统发送心跳包并获取数据刷新页面。前端用户系统发送消息使用 POST 方式提交至后端 Nodejs 服务器，Nodejs 服务器进行用户数据处理与数据库访问。

4 结束语

该网站目前对 Flutter 框架进行了深度应用，系统绝大部分代码由 Dart 语言组成，大量调用 Flutter 框架，验证了 Flutter 技术对于实际应用的可行性。但在开发过程中，有对于不同平台的功能实现阻碍，代码在不同平台上表现不一致，这是由于 Flutter Web 渲染的部分局限性造成的。

项目文件总体过大，因此此处仅包含工具压缩构建后的网页，项目主体代码已在 Github 上进行开源，开源地址为 https://github.com/Ricardo2001ZG/bu_yu_xi_yan。为验证所有权，本论文已经同步上传至项目下的 PRD 文件夹。

后端采用 Nodejs 服务器推送 Json 数据的方式对前端页面进行更新，在 5MB 的期末考核限制下无法部署动态服务，导致此处仅能展示静态测试数据页面。

演示页面对前端系统动态部分进行了裁剪，采用了静态测试数据进行页面展示。

由于 Flutter Web 页面为动态渲染，且测试数据使用本地文件图片，必须使用服务端程序部署才能够正常浏览，因此项目中附带了 Nginx 服务端，需在全英文路径下才能正常启动。