



华侨大学

HUAQIAO UNIVERSITY

本科毕业论文（设计）

题目：宠宝平台设计与实现——前端
集成与后端开发

学 院	计算机科学与技术学院
专 业	软件工程
年 级	2020级
学 号	2025123051
姓 名	林子旺
指导老师	萧宗志、副教授

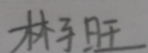
华侨大学教务处印制

2024年6月

华侨大学本科毕业论文（设计）诚信承诺书

本人（姓名）林子旺 学号 2025123051
专业 软件工程 郑重承诺：所呈交的毕业论文（设计）
（题目：宠宝平台设计与实现——前端集成与后端开发）是在
指导教师指导下自主完成，毕业论文（设计）选题和研究内容
中不存在不正当引用、抄袭、伪造、篡改、代写、买卖等行为，
如有违规行为，本人愿意承担一切责任。

承诺人（签名）：



2024 年 06 月

摘要

现今，人们都饲养起了宠物来满足自己的情感需求。饲养宠物已成为一种流行的趋势。然而许多人并未意识到饲养宠物需要付出大量的精力，时间，金钱和责任。这导致了許多宠物被弃养的现象，街上流浪宠物数量激增，外来物种入侵，破坏了当地的生态平衡，等带来了一系列问题。

因此，在这一系列问题中，本文得到了启发并设计了一款宠物试养平台——《宠宝》，希望借由此软件平台来减轻甚至解决此类问题。本软件平台拥有一个特点也就是试养。试养的概念是指用户再不确定是否适合饲养宠物，是否具备充分条件和环境来饲养宠物的情况下，可以领养一段时间某一种类的宠物来提前准备或者适应，以使用户判断否真的适合饲养宠物。这有点类似于租借的概念。当然本软件平台的用户也可以在试养阶段后选择是否领养该宠物。

除了解决弃养宠物的问题，《宠宝》系统还拓展了许多功能，例如宠物医院、商城、宠物寄宿、聊天等，以满足用户一站式的需求。从而达到减轻用户即饲主的负担和压力，更加享受饲养宠物所带来的乐趣。本软件平台的前端采用 Flutter 框架设计，提供跨平台的 UI 界面，使用 Dart 作为编程语言；后端采用 Spring Boot 处理数据库访问的 APIs，数据库使用 MySQL。

关键词：宠物试养；宠物试养平台；Flutter 框架；Dart 语言；SpringBoot；Mysql 数据库

Abstract

Nowadays, people keep pets to satisfy their emotional needs. Keeping pets has become a popular trend. However, many people do not realize that raising a pet requires a lot of energy, time, money and responsibility. This has led to the abandonment of many pets, a surge in the number of stray pets on the streets, and the invasion of alien species, which has destroyed the local ecological balance and has brought about a series of problems.

Therefore, in this series of problems, this article was inspired and designed a pet trial breeding platform-"Pet Baby", hoping to use this software platform to alleviate or even solve such problems. One feature of this software platform is trial breeding. The concept of trial breeding means that when users are no longer sure whether they are suitable for raising pets and whether they have sufficient conditions and environment to raise pets, they can adopt a certain type of pet for a period of time to prepare or adapt in advance, so that users can judge whether they are really suitable for raising pets. pet. This is somewhat similar to the concept of leasing. Of course, users of this software platform can also choose whether to adopt the pet after the trial adoption stage.

In addition to solving the problem of abandoned pets, the "Pet Baby" system has also expanded many functions, such as pet hospitals, shopping malls, pet boarding, chat, etc., to meet users' one-stop needs. In this way, the burden and pressure of the user, that is, the owner, can be reduced, and the user can enjoy the fun of raising pets more. The front-end of this software platform is designed using the Flutter framework, providing a cross-platform UI interface, and uses Dart as the programming language; the back-end uses Spring Boot to handle APIs for database access, and the database uses MySQL.

Keywords: Pet trial breeding; Pet trial breeding platform; Flutter framework; Dart language; SpringBoot; Mysql database

目录

摘要	I
Abstract	III
第一章 绪论	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 选题意义	2
1.2 国内外研究现状	3
1.3 论文的主要工作	5
1.4 本章小结	6
第二章 相关工具及技术平台	7
2.1 C/S（Client-Server）结构	7
2.2 Flutter 框架以及 Dart 编译语言	7
2.2.1 Flutter 框架	7
2.2.2 Dart 编译语言	11
2.3 SpringBoot 框架与 RESTFUL APIs	12
2.3.1 SpringBoot 框架	12
2.3.2 RESTFUL APIs	12
2.4 MySQL 数据库	12
2.5 Android Studio IDE	12
2.6 本章小结	13
第三章 需求分析	15
3.1 需求综述	15
3.2 可行性研究	15
3.2.1 技术可行性	15
3.2.2 经济可行性	16
3.2.3 时间可行性	16
3.3 软件开发模型的采用	17

3.3 平台功能模块设计	18
3.4 平台性能需求分析	19
3.5 平台界面设计及风格	20
3.6 本章小结	21
第四章 系统设计	23
4.1 平台用例图设计	23
4.2 功能模块划分及说明	24
4.3 平台 ER 图以及数据库表设计	26
4.4 本章小结	36
第五章 功能开发	37
5.1 各界面时序图与流程图	37
5.2 前端实现与开发	46
5.3 后端数据库与接口	50
5.4 本章小结	52
第六章 运行与测试	53
6.1 功能与性能测试	53
6.2 成果展示	55
6.3 平台安装部署及步骤	73
6.4 本章小结	75
第七章 总结和展望	77
参考文献	79
致谢	81

第一章 绪论

1.1 研究背景及意义

1.1.1 研究背景

随着社会的蓬勃发展，经济的持续增长不仅提升了人们的生活水平，也激发了对情感需求的关注^[1]。在满足了马斯洛的需求层次理论中的生理和安全需求后，人们逐渐将目光转向社交需求，这促使许多人选择饲养宠物来填补内心的空缺。因此，饲养宠物已经成为一种时尚潮流。不分年龄，无论老幼，人们都希望有一只可爱的伙伴陪伴身边。如今，人们也可以根据个人的兴趣和生活方式，提供给人们选择的宠物的种类多种多样，例如：狗、猫、小型哺乳动物、鸟类、爬行动物以及水族动物等。不同种类的宠物拥有各自独特的特点。

然而，许多人并未充分认识到饲养宠物所需投入的巨大精力、时间、金钱和责任^[2]。这一点尤其在小孩们身上显得格外突出。由于父母工作繁忙，孩子们常感缺少陪伴，因而在一时冲动、好奇心和渴望新鲜感的驱使下，孩子们可能会渴望拥有一只宠物。然而，随着时间的流逝，新鲜感逐渐褪去，孩子们可能会发现养宠物带来的责任过于繁重，于是将所饲养的宠物束之高阁。最终，这些宠物可能会被遗弃。此外，由于缺乏经验和了解，许多饲主不知如何正确饲养和训练宠物，导致宠物可能逃跑、捣乱甚至与打扰周围邻居。随着耐心的消磨，饲主可能会选择放弃宠物^[3]。另一个常见情况是，饲主可能会突然改变生活环境，或者意识到当前的环境并不适合某种类型的宠物。在这种情况下，饲主可能会选择将宠物遗弃。

以上种种现象都是把宠物弃养的常见原因。这也间接导致城市中流浪宠物数量的激增，外来物种的大量入侵，从而破坏了当地的生态平衡。流浪宠物无处可归，不仅面临着饥饿和疾病的威胁，还可能成为城市环境的问题源。流浪宠物会扰乱垃圾桶、传播疾病，甚至攻击行人或其他动物。同时，有些被遗弃的宠物是外来物种不属于当地，这些物种会对当地生态系统造成严重破坏。流

浪宠物会竞争当地物种的食物资源，破坏当地植被，甚至传播疾病。这种对生态平衡的侵害可能会影响到整个生态系统的稳定性，给生态环境带来长期而严重的负面影响。

接着，即使饲主做足了充分的准备和关爱，饲养宠物的过程中仍然会面临许多困扰。其中一个常见问题是，饲主不得不长时间离家处理紧急事务的这段时间内，照顾宠物就成为了一大难题。即使选择将宠物送至宠物寄宿中心，饲主也必须确保选定的地方能够提供高质量的照顾和环境，以避免遭遇不良商家。另外，当宠物生病时，饲主也会面临着诸多挑战^[4]。在这紧急情况下，饲主需要迅速找到附近的宠物医院，而且还要确保这些医院拥有合格的资质和设施，以保障宠物得到及时的治疗。除了治疗，宠物还需要定期的健康检查和疫苗注射才能够确保宠物的身体健康。饲主需要常常和这些机构打交道，大大增加了饲主的不便性。

1.1.2 选题意义

由此可见，饲养宠物并非易事，需要全方位考虑。因此，本文便从中获取灵感并设计与实现一款全面、高效的宠物试养平台——《宠宝》。《宠宝》旨在更好地满足饲主的需求，提升饲主在宠物抚养过程中的便捷性和质量。本软件平台可以让饲主更轻松地管理宠物的需求，确保宠物在各种情况下都能得到适当的关爱和照顾，让饲主在宠物抚养过程中更加愉快。

首先，《宠宝》的一个重要功能是能够进行宠物试养。用户可以通过《宠宝》平台试养宠物一段时间，以确定这种类型的宠物是否适合自己，并确保自己有能力承担养宠的责任，而非仅仅是一时的冲动或好奇心。同时，饲主也可以通过试养体验不同品种的宠物，从中获得丰富的乐趣和经验。这样一来，可以大大降低宠物被遗弃的现象，从而解决了上一节所提到的因弃养导致的一系列社会、环境和生态问题。

此外，宠宝具有宠物寄宿的功能，就如上一节所述，饲主们在一段时间内无法亲自照料宠物或无法及时找到可信赖的寄养中心时，可以通过平台查询附

近认证的寄养宠物店家，确保宠物得到良好的照顾。此举不仅方便了饲主，也保障了宠物在主人离开期间的舒适和安全。《宠宝》还整合了宠物医院的功能，让饲主能够轻松查找附近持有认证的宠物医院，确保宠物能够获得及时和专业的医疗服务。此外，平台还引入了 AI 宠物医生助手功能，饲主可以通过这个功能咨询关于宠物异常行为的问题，或者在 AI 医生的指导下采取一些简单的措施，并获取宠物健康相关的知识。

此外，《宠宝》还具备了一系列额外功能，旨在协助饲主解决在饲养宠物过程中遇到的各种问题。这些功能包括商城，让饲主能够轻松订购宠物的日常用品；宠物美容服务功能，让饲主轻松找到附近满意的宠物美容店来为宠物洗澡、理毛等；还有邮箱功能，让饲主可以与其他饲主分享彼此的宠物日常，并接收来自店家或系统的消息。此外，平台还具备推送一些精选帖子和视频的功能，供饲主学习和收藏，以提升养宠技能。当然，饲主也可以在平台上自行发帖，分享自己的宠物经历和心得。

总的来说，《宠宝》是一款致力于解决饲主在宠物抚养过程中所遇到问题的全方位平台。它提供了一系列集成功能，旨在减轻饲主的负担，为他们提供便利。其中，最突出的是试养功能，能够显著降低宠物被遗弃的现象，为此类问题提供了有效的解决方案。此外，《宠宝》也为相关产业带来商机，推动了宠物产业的健康发展。

1.2 国内外研究现状

在近年来宠物市场在中国国内都在不断成长且非常庞大。再加上，科技的发展，人人有手机后，人们的生活起居都希望能靠一机解决，所以对于宠物管理相关的应用软件的需求越来越大了。并且，对其功能的需求也要越来越多样化。像是能够发布和评论帖子，加好友交流，刷短视频等。还有最重要的一点就是用户的数据，隐私和安全必须收到保护。对比国外，宠物管理相关软件就是越来越智能化，能够配备硬件组合例如智能颈环、摄像头等来管理宠物和更了解宠物的习性。此外，就是宠物管理相关软件能够跨平台且兼容，例如支持

IOS, Android, 网页等。还有就是能够远程医疗, 智能助手医疗了。通过软件实时监测宠物的健康状况, 并与兽医进行在线沟通。总的来说, 现在人们对于宠物管理软件的要求就是能够功能更多样性, 更智能化, 用户数据隐私安全受到保障, 能够一机解决, 软件能够跨平台且兼容, 能够实现远程照看, 治疗等。如下图图 1.1 是本文对于市场上存在的各类宠物软件平台相关的进行调查并总结出来各类宠物软件平台的优缺点以及功能以方便对比和作为本文开发本平台的参考与借鉴。

维度	E 宠	波奇宠物	铲屎官的日常	句苗岛
定位	电商	电商+社交	电商+社交	020+电商
应用分类	购物	购物	生活	生活
用户注册和登录	✓	✓	✓	✓
宠物试养	×	×	×	×
宠物医院	✓	×	×	×
宠物商城	✓	✓	✓	✓
宠物分享	✓	✓	✓	✓
宠物寄养	×	×	×	✓
宠物美容	×	×	×	×
通知功能	✓	✓	✓	✓
支付功能	✓	✓	✓	✓
宠物训练课程	✓	×	×	×
社交功能	×	✓	✓	✓
第三方服务(商家)入驻平台	×	×	×	×
我的钱包	✓	✓	×	×
客服	✓	✓	✓	✓

图 1.1 各类宠物软件平台的差异

在研究完国内外对于此议题的对比后, 本文还进行对软件开发的工具平台进行了国内外研究。本文选择了近几年火热的开发工具 Flutter 框架进行研究。Flutter 框架是前端 UI 工具包而后端则是 JAVA 编程语言的 SpringBoot 框架处理数据库访问和数据库是采用 Mysql。Flutter 是一种由 Google 开发的开源 UI 工

具包。Flutter 在移动端应用软件开发领域里非常流行^[5]。根据 statista(全球性的在线统计数据平台)统计全球软件开发员使用的移动端框架从 2019 年的 30%提升到了 2022 年的 46%，临近一半的软件开发员都在使用 Flutter 框架。Flutter 框架也被很多大企业使用，像是 Google classroom, 阿里巴巴集团，字节跳动，腾讯等都在使用 Flutter 开发软件。而 Flutter 的特点就是跨平台开发，使用同一个代码库可以在 IOS，WEB，Android 等平台使用。Flutter 还有一个闪光点就是支持热重载开发者可以在运行应用的过程中即时查看代码变更的效果，极大地加快了开发迭代的速度。再补充的是 Flutter 拥有丰富的预置组件库，称为 Widgets，覆盖了各种 UI 元素和布局。简言之，介于以上特点就是 Flutter 在中国内外和未来的前景都是非常的好。接着，SpringBoot 是基于 Spring 框架的一个开发框架。使用 Spring Boot 框架来处理数据库访问，以确保数据的一致性和安全性^[6]。

总的来说，通过结合 Flutter 和 Spring Boot 的优势，本文打造了一个功能丰富、智能化且安全可靠的软件，满足现代人对于宠物管理的多样化需求。同时，利用 Flutter 的跨平台特性，使得这样的软件能够在全球范围内广泛应用，具有良好的发展前景。

1.3 论文的主要工作

在本文中，本文基于 C/S 结构，采用 Android Studio 作为 IDE，利用 Flutter UI 框架和 Dart 编程语言进行开发，结合后端技术如 Spring Boot 和 MySQL 数据库，完成了一款名为《宠宝》的宠物试养平台的全套开发流程。首先，本文对整个系统进行了需求分析，明确了用户需求和功能模块，区分了用户群体如用户、店家、管理员。在需求分析的基础上，进行了概要设计，设计了系统的整体架构和模块划分，确定了前后端的通信方式和数据传输格式。接着，本文进行了详细设计，对每个模块进行了深入的设计和实现，包括界面设计、功能实现、数据交互等方面。随后，本文进行了运行测试，测试了系统的稳定性以及功能。最终，本文实现了《宠宝》平台的界面。

本文几个部分：

第一章：主要介绍了研究背景和意义。并分析了宠物相关平台的国内外研究现状分析。

第二章：介绍本文所用到的技术平台与相关路线。以 C/S 结构，Flutter 框架，MySQL 数据库进行宠物平台的设计与实现。让用户可以通过此平台解决在饲养宠物的过程中遇到的一些繁杂的事物以及减缓弃养宠物现象。

第三章：分析介绍系统的需求、进行了可行性研究、分析平台功能、区分功能模块、性能需求分析、工作流程、数据库设计、界面风格设计。

第四章：介绍宠物试养平台各个模块之间的联系和关系。

第五章：讲解如何开发本软件平台。

第六章：对已设计的软件平台进行测试、运行以及安装部署。

第七章：总结和展望。

1.4 本章小结

本文首先讲解了饲养宠物成为了一种流行趋势，因许许多多的原因而导致了弃养宠物的现象发生，并引发了许多问题。为了减缓甚至解决此问题所开发了一款宠物试养手机应用平台。接着，也扩展了此平台的功能来帮助饲主，饲主能够通过此平，获得一站式的服务，减轻了饲主的负担。接着，也介绍了使 Flutter+Dart+SpringBoots+Mysql 的全栈开发来实现此平台。

第二章 相关工具及技术平台

在当今社会科技的快速发展中，软件开发领域涌现了众多技术和平台，每一种都各具特色，各有优缺点。根据不同的使用场景和需求，选择合适的技术工具能够事半功倍，使开发过程更加高效、灵活，并且能够更好地满足用户的需求。以下是本文使用的技术平台以及工具。

2.1 C/S (Client-Server) 结构

C/S(Client-Server)结构是一种将客户端与服务器分割开来的分布式架构。每一个客户端软件的实例都可以向一个服务器或应用程序服务器发出请求，再由服务端返回结果即响应。如今大部分的手机应用软件都采用 C/S 结构。因此本系统也采用 C/S 结构。

图 2.1 展示了 C/S 工作流程的精髓。透过图示，可以直观地理解其工作原理。简而言之，用户通过客户端向服务器发送请求，以获取诸如数据库访问等资源。而服务端在处理完毕后，则会将响应传递回客户端，实现了双向交互。

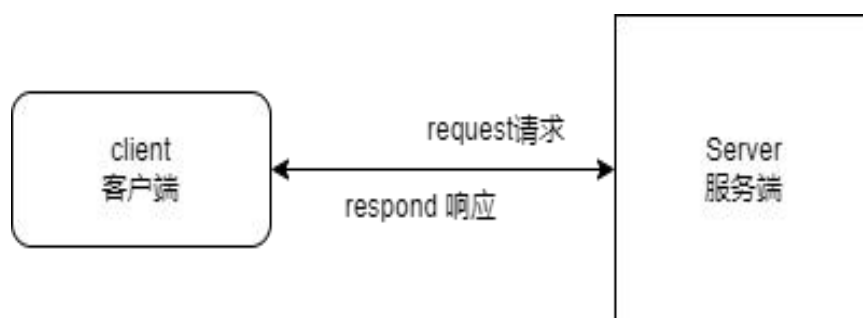


图 2.1 C/S 工作流程图

2.2 Flutter 框架以及 Dart 编译语言

2.2.1 Flutter 框架

Flutter 是一款由 Google 开发的开源 UI 工具包，旨在构建跨平台的移动应用程序。自其 2018 年发布以来，Flutter 已经成为移动应用开发领域的热门选择。

许多企业和应用软件开发公司已相继转换成使用 Flutter 开发。此外 Flutter 是完全免费的、开源的和具备了以下许多优点：

1. 跨平台性：Flutter 具备无与伦比的跨平台性。开发人员可以借助单一代码库轻松构建同时运行于 iOS 和 Android 两大主流移动端操作系统的应用外也可以在 Web 上运行。而且，不同平台上 Flutter 的 UI 风格和交互行为保持一致，为用户提供了无缝的体验。
2. 快速开发：Flutter 具备热重载功能，其独有的热重载功能极大地缩短了开发周期。开发者可以在应用程序运行的同时即时查看代码修改的效果，无需重新启动应用程序。这使得开发过程更加高效，让开发者能够快速迭代并实现所需功能。
3. 丰富的功能支持：Flutter 框架提供了丰富的功能支持，包括网络请求、数据库操作、状态管理、国际化、测试等方面的功能，使开发人员能够轻松构建功能完备的应用程序。
4. 灵活的 UI：Flutter 提供丰富的内置 UI 组件和自定义 UI 组件库，开发人员可以轻松创建漂亮、流畅的用户界面，并实现各种复杂的 UI 交互效果。此外，再加上 Flutter 使用自带的渲染引擎 Skia，可以直接绘制 UI 组件而无需依赖平台原生控件，因此具有优秀的性能表现，能够实现高达 60fps 的流畅动画效果^[7]。

Flutter 之所以具有诸多优势，归功于其独特的特色与一致的统一对象模型：widget。在 Flutter 中，widget 扮演着至关重要的角色，图 2.2 展现了这种结构。每个 widget 嵌入其中，并继承其父项的属性。不同于其他框架，Flutter 没有单独的“应用程序”对象，而是由根 widget 来承担这个角色。开发人员可以通过告诉框架使用另一个 widget 替换层次结构中的 widget 来响应事件，例如用户交互。替换后，框架会比较新的和旧的 widget，并高效地更新用户界面。

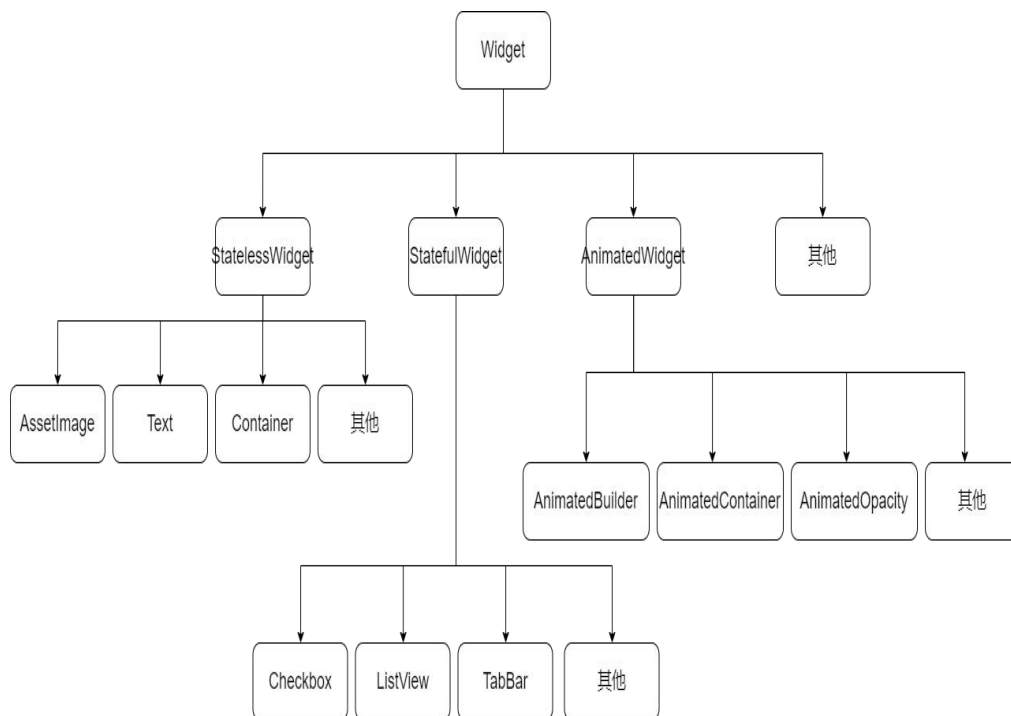


图 2.2 Widget 对象模型层次结构图

Widget 本身通常由许多更小的、单一用途的 widget 组成，这些 widget 结合起来产生强大的效果。例如，Container 是一个常用的 widget，Container 继承了 StatelessWidget 再由多个 widget 组成，负责布局、绘制、定位和调整大小。具体来说，Container 由 LimitedBox、ConstrainedBox、Align、Padding、DecoratedBox 和 Transform 等组件构成。开发人员可以以各种方式组合这些以及其他简单的 widget，而不是继承容器。Flutter 的类层次结构非常浅且宽，这使得最大限度地增加了可能的组合数量。此外，开发人员还可以通过与其他 widget 组合来控制 widget 的布局。例如，要将 widget 居中，可以将其封装在 Center widget 中。填充、对齐、行、列和网格等布局 widget 没有自己的可视化表示，它们的唯一目的是控制另一个 widget 的布局某些方面。

图 2.3 就展示了 Flutter 是个扩展的分层系统，由一系列独立的库组成，每一个库都依赖于低层。所以上层比下层使用频率更高。此设计的好处就是帮助开发人员用更少的代码就能实现更多的东西。其架构分为三层：

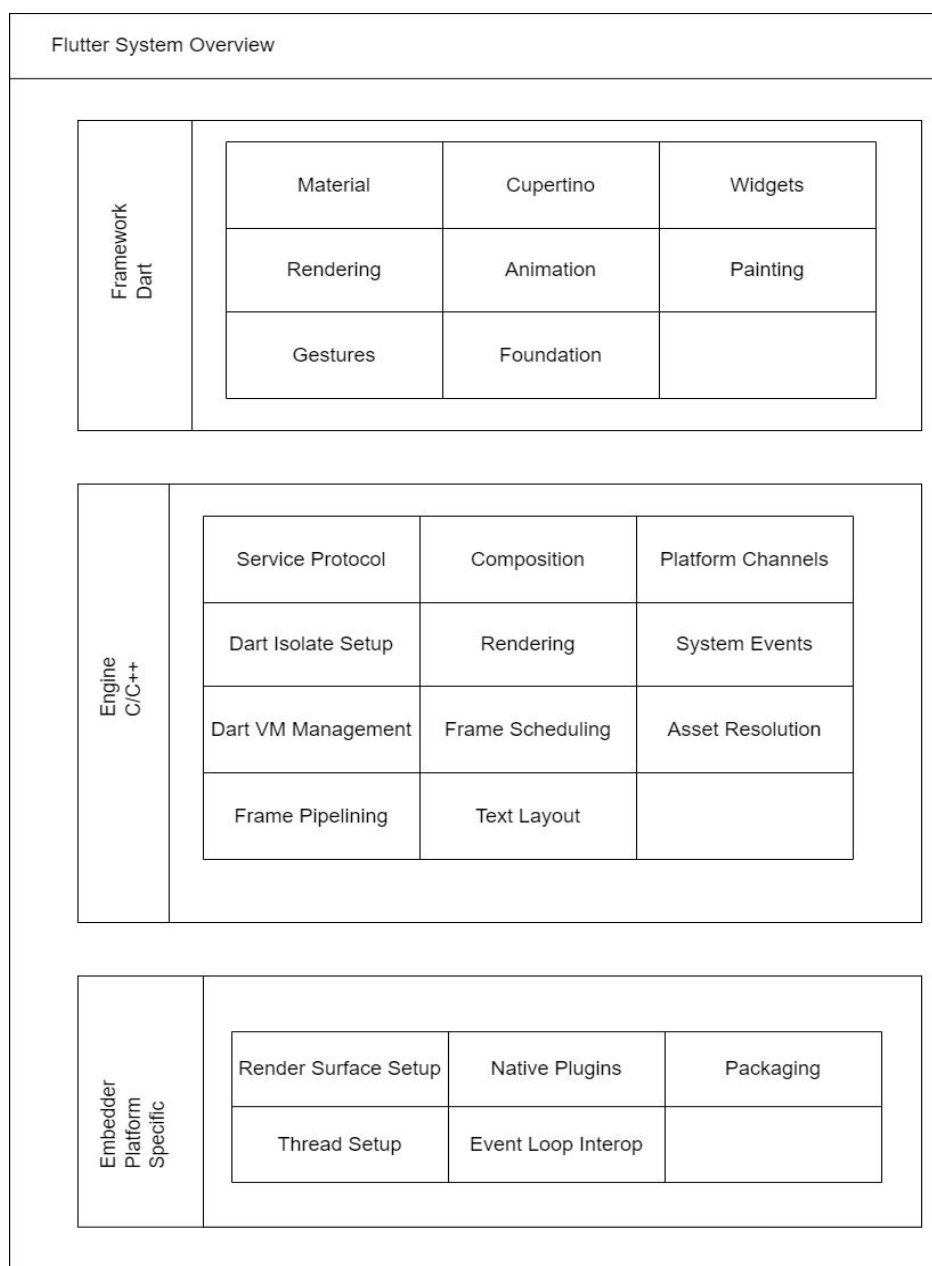


图 2.3 Flutter 的架构图

1. Embedder 位于最底层，是 Flutter 嵌入到任何平台的适配器，提供了与底层操作系统的通信入口点，以及管理消息事件循环的功能^[8]。
2. Engine 是 Flutter 的核心，用 C++编写，支持所有 Flutter 应用程序提供所必需的原语。它负责对合成场景进行栅格化，提供了低级实现，包

括图形、文本布局、文件和网络 I/O、可访问性支持、插件架构以及 Dart 运行时和编译工具链^[9]。

3. Framework 位于最上层，由引擎通过 `dart:ui` 暴露，将基础 C++ 代码包装在 Dart 类中。Framework 提供了一种以 Dart 语言编写的现代、反应式架构，包括了一系列丰富的平台、布局和基础库。从底部到顶部，Framework 提供了一组丰富的平台、布局和基础库，包括了 Foundational 基建类、Rendering 渲染层、Widgets 小部件层以及 Material 和 Cupertino 库。

2.2.2 Dart 编译语言

Dart 是一种由 Google 开发的面向对象编程语言，其语法类似于 C 语言，被广泛应用于 Flutter 框架的开发中。Dart 具有以下几个显著特点：

1. 直观的语法：Dart 具有直观易懂的语法，类似于 Java 和 JavaScript，使得学习和使用变得更加容易。
2. 即时编译：Dart 支持即时编译（JIT）和预编译（AOT）两种编译模式，这使得开发过程更加迅速高效，同时在执行时也能保持高性能。
3. 健全的空值安全：Dart 内置了强大的空值安全特性，只有在显式声明变量可以为空的情况下，该变量才可以为 `null`。通过静态代码分析，Dart 在运行时可以有效地保护开发者免受空值异常的影响，确保代码的稳健性和可靠性。
4. 丰富的核心库和 APIs：Dart 拥有丰富的核心库和 APIs，同时也有大量的第三方发布者和社区发布的软件包，为开发者提供了丰富的工具和资源，方便开发各种类型的应用程序。

2.3 SpringBoot 框架与 RESTFUL APIs

2.3.1 SpringBoot 框架

Spring Boot 是一个专注于快速构建基于 Spring 框架的应用程序的开发框架。它大大简化了 Spring 应用程序的配置和部署过程，提供了丰富而强大的开箱即用的功能，如自动配置、内嵌式服务器和监控等。

本文利用了 Spring Boot 的便利特性，集成了 Spring 框架中的 Spring MVC 模块来构建 RESTful APIs。

2.3.2 RESTFUL APIs

RESTful APIs 是一种基于 REST（Representational State Transfer）架构风格的 API 设计规范。它使用 HTTP 协议的各种方法（GET、POST、PUT、DELETE 等）来实现对资源的增删改查操作，并通过 URL 来唯一标识资源。本文将利用 RESTful APIs 技术来访问数据库，通过符合 RESTful 设计规范的 API 接口来实现对数据库资源的管理和操作，从而实现高效、灵活的数据交互和处理。

2.4 MySQL 数据库

MySQL 是一种关系型数据库管理系统，广泛用于 Web 应用程序和移动应用程序的数据存储和管理^[10]。它具有高性能、可靠性和扩展性，支持复杂的数据操作和查询，适用于各种规模的应用场景。MySQL 数据库被用作本文的数据存储和管理工具，以确保数据的安全性和一致性^[11]。

2.5 Android Studio IDE

Android Studio 是一款由 Google 推出的集成开发环境（IDE）^[12]，专门用于 Android 应用程序的开发。作为 Android 开发的主要工具之一，Android Studio 提供了丰富的工具和功能，旨在帮助开发人员更轻松、高效地创建、测试和调试 Android 应用程序。

Android Studio 集成了诸多强大的特性，包括代码编辑器、虚拟设备管理器、调试器、性能分析器等，这些功能使得开发人员可以在一个统一的环境中完成整个开发过程。本文选择使用 Android Studio 作为开发工具的原因之一。是其强大的 Android 虚拟设备功能（AVD），这允许开发人员在本地模拟不同的 Android 设备，从而更方便地进行应用程序的测试和调试。

2.6 本章小结

本节主要介绍了本文中所用来开发平台《宠宝》的框架、工具及技术。主要分为前端(Flutter UI 框架和 Dart 编译语言)，后端(SpringBoot 和 MySQL)以及 IDE Android Studio。

第三章 需求分析

3.1 需求综述

通过第一章的所讲述的课题背景和意义，可以看出项目的问题定义，第二章的相关工具及技术平台的使用则是给出了本软件平台可以使用的工具及技术。因此，本章将对《宠宝》平台的可行性进行研究，接着再对其需求详细分析，包括功能性需求,非功能性需求,采用哪个软件工程过程模型框架。以确保系统设计与实现能够更加快速，稳定并达到预期的效果。

3.2 可行性研究

可行性研究是用于评估本软件平台在各种方面的可行性和潜在成功率。其目的是确定本软件平台是否值得继续开发，并识别出可能影响其成功的各种因素。以下是本文对本软件平台进行关于可行性研究的主要部分：

3.2.1 技术可行性

技术可行性评估项目在技术层面上的可行性，就如第二章节所述本文已确定了该使用哪些工具及平台来实现本软件平台。以上所提及的工具和技术平台都具有广大及非常活跃的社区活跃度以及开源。因此本文在开发本软件平台的过程中，如有遇到任何技术上的问题及困扰都能在该技术平台的社区里寻求帮助。许多开发者都会在对应的技术社区里讨论及分享所遇到的问题并进行讨论。从中，也体现了本文所使用工具和技术平台已非常成熟并广泛被使用。此外，在 2.1 节里本文已对市场上类似于本软件平台的软件进行了许多的特点分析。因此，已有了成熟的解决方案，本文可以从中借鉴其特点以及不足，并改进到本软件平台上。

总之，在技术可行性方面，本文已确保了本软件平台的技术是可行的。本软件平台的技术资源都非常到位。

3.2.2 经济可行性

经济可行性评估本软件平台开发的经济合理性和成本效益。在成本估算这方面，本文使用的工具以及技术平台均为免费使用。因此，本软件平台的开发成本只有人力成本也能视为时间成本。由于，上节所述，本文使用的工具及开发平台都具备开源，所以减少了本文的开发时间。而在硬件部分，本软件平台的开发不需要很高配置的笔记本、主机或其他设施，因此本文的现有的硬件环境具备开发本软件平台的条件。

总之，本软件平台的开发成本是非常低的且本问能够在没有外来投资的情况下完成本软件平台。所以本软件平台的经济可行性是可行的。

3.2.3 时间可行性

时间可行性评估开发本软件平台的时间安排，确保本软件平台能在预定时间内完成。以下表 3.1 为本文开发软件平台的时间安排：

表 3.1 开发软件平台的时间表

过程	时间
分析需求和查阅参考文献	2023-10-30 至 2024-01-07
了解 APP 开发原理	2024-02-26 至 2024-03-01
构思主题初期策划，分析需求等	2024-03-12 至 2024 -03-16
构建 APP 前端	2024-03-16 至 2024 -03-30
构建 APP 后端	2024-03-16 至 2024-03-26

表 3.1（续）

过程	时间
集成 APP 前后端	2024-03-26 至 2024-04-06
中期检查，分析，改进	2024-04-06 至 2024-04-15
App 功能与性能测试以及改进	2024-04-16 至 2024-04-30
论文撰写	2024-05-01 至 2024-05-07
送审，修改论文，准备答辩 PPT	2024-05-13
论文撰写修改完成	2024-05-27

如表所示，本软件平台的开发是执行前后端并行开发的。如此，能大大减少了本软件平台的开发周期，不需要等待前端开发完后才开发后端。因此，本软件平台按照表 3.1 的时间表能在给与的时间完成开发，时间可行性是可行的。

3.3 软件开发模型的采用

软件开发模型是为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。因此，为了更加有序地完成本文的软件平台，本文采用的软件工程框架模型是快速原型模型。图 3.1 就展示了快速原型模型框架图。

快速原型是快速建立起来的可以在计算机上运行的程序，所能完成的功能往往是最终产品能完成的功能的一个子集。本文采用快速原型模型是因其特点

能快速开发项目，开发周期短，节约成本。快速原型模型适用于需求不明确或容易变化的项目。因此本文可以在开发软件平台的过程中，先开发一个简易的软件平台以其为原型在不断地该并完善本软件平台。此外，本文也能在开发过程中不断调整开发方向，灵活应对需求变化，使得本文能够在本软件平台开发中不断试错，最终实现高质量的软件平台开发目标。

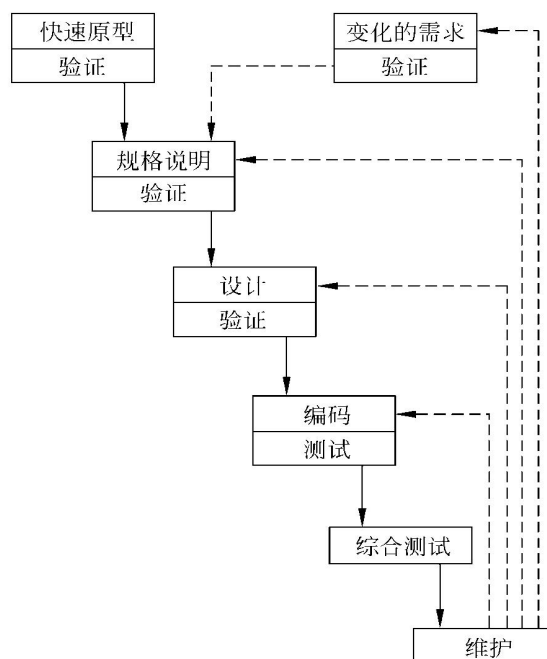


图 3.1 快速原型模型框架图

3.3 平台功能模块设计

在第一章的选课背景和意义部分，本文分析出了用户对于《宠宝》平台的期望以及需求，并对该客户需求功能的基本下，扩充了其他功能。

1. 用户账户功能：包括用户注册、登录、个人信息管理等功能，以便用户使用本平台软件系统《宠宝》。
2. 试养宠物功能：该功能使用户能在平台上选择并试养他们感兴趣的宠物，包括查看相关信息、预约宠物、选择试养时长等。这一功能是平

台的核心，通过试养概念，帮助用户确认适合他们的宠物类型，并提供新的体验。

3. 寄宿宠物功能：用户可以查询并预约附近的宠物寄宿店家，选择所需的寄宿时长，查看寄宿中心的环境等，这项功能解决了用户需要出远门时对宠物照顾的问题。
4. 宠物医院功能：允许用户查询附近的宠物医院，并预约就诊服务。用户可以查看医院提供的服务，如注射疫苗、身体检查、微型手术等，从而节省排队等待的时间。
5. AI 宠物助手功能：提供智能助手服务，向用户提供饲养宠物的常识和训练等信息。这有助于用户更好地了解各种宠物的习性和行为，以及在宠物出现异常情况时采取预防措施，减少用户的困惑。
6. 商城功能：平台提供各种宠物日常用品、玩具、服装等商品，为用户提供便利的购物体验。
7. 宠物美容服务功能：用户可以预约宠物洗澡、理发、美容护理等服务，让用户放心，宠物更加美丽，尤其对于大型毛绒宠物而言。
8. 邮箱功能：用户可以及时收到来自商家、系统等的信息，并与他们交流，扩大社区内的朋友圈。
9. 精选帖子和视频功能：提供优质的宠物相关帖子和视频，供用户学习和娱乐。

3.4 平台性能需求分析

一个有质量的软件除了满足用户的功能性需求外，还需要考虑到性能需求包括效率、安全、可用性等方面的要求，一般用户不会明确指出，但是却需要开发人员自行添加，否则会影响用户的体验。

1. 效率要求：平台需要具有良好的性能，能够快速响应用户的请求，保证系统的稳定性和可靠性。

2. 安全要求：用户的个人信息和交易数据需要得到保护，平台需要具备数据加密、身份验证、访问控制等安全机制。
3. 可用性要求：平台需要具备良好的用户界面设计和用户体验，方便用户进行操作和交互。
4. 跨平台兼容性：平台需要具备跨平台兼容性，能够在不同的设备和操作系统上正常运行和使用。

3.5 平台界面设计及风格

一个软件平台要让用户有个深刻的印象和要吸引新的用户，其界面风格设计占据着非常重要的地位。想要设计良好的界面首先要分析用户类型。用户分成偶然性、生疏型、熟练型和专家型。本文的用户类型多为偶然型用户。偶然型用户是第一次使用系统的或者是极少使用系统的用户。这样的用户，对系统完全没有记忆，每次使用都要经历一次重新学习的过程。对于这类用户，界面的功能必须简单，所有的操作，必须能预防错误，能及时给予用户提示和反馈信息。

因此，本文运用界面设计的黄金规则即界面设计的基本定律，是多位前人从实践中总结出的一些设计原则。

1. 努力做到一致性：在操作平台时，操作环境都是一致性的，界面的布局，背景颜色，字体等都要始终保持一致。
2. 允许熟练用户使用快捷键：设有缩略语、特殊按键、隐含的命令和宏以减少用户的交互次数，提高与用户的交互速度。
3. 价值的反馈：当用户操作平台时添加简短的反馈信息，而对于不常且重要的操作则反馈信息丰富多一些。
4. 设计说明对话框以生成结束信息：一组操作结束后设有反馈信息，这可以使操作者产生完成任务的满足感和轻松感，而且。可以让用户放弃临时的计划和想法，并告诉用户，系统已经准备好接受下一组操作。

5. 提供预防错误和简单的错误处理手段：尽可能不让用户犯严重的错误。可采取某些措施，如使用菜单选择风格。而不是表格填充风格，不允许在数字输入字段存在字母。如果用户犯了错误，系统应能检测到错误，并给出简单、积极、具体的批示。以使用户恢复正常。
6. 允许轻松的反向操作，操作应尽可能地允许反向。
7. 支持内部控制点：用户非常希望能控制系统，并希望系统对他们的操作进行反馈。
8. 较少短时记忆：多页显示统一和窗口移动频率低，并且要保证分配足够的时间用于学习代码、记忆方法操作序列

3.6 本章小结

本章以第一章的选题背景及意义进行了需求分析。对《宠宝》平台的设计有了“必须做什么”的概念，可以为后续的设计和实现工作提供重要参考，确保平台能够满足用户的需求并达到预期的效果。其中包括功能性需求和非功能性需求，应该用什么过程模型来进行本文的项目和应该设计出怎样的人机界面。

第四章 系统设计

系统设计就是以前一章需求分析为基础，开发人员要“怎样做”才能实现整个平台。因此这个章节就是主要介绍该怎样去实现整个平台系统。本章会用统一建模语言 UML 来协助阐明，构建本平台系统。UML 的好处在于可以以很简单的方式介绍系统的结构，并且读者页非常容易读懂，不需要知道太多的专业用语。

4.1 平台用例图设计

图 4.1 是以用例图展示了本文的软件平台系统《宠宝》的整个总框架。如图所示，参与者是用户，用户通过参与系统具备的用例，然后用例之间又存在着包含和扩展的关系。用户就是通过这些关系来使用各个用例。例如图里的用

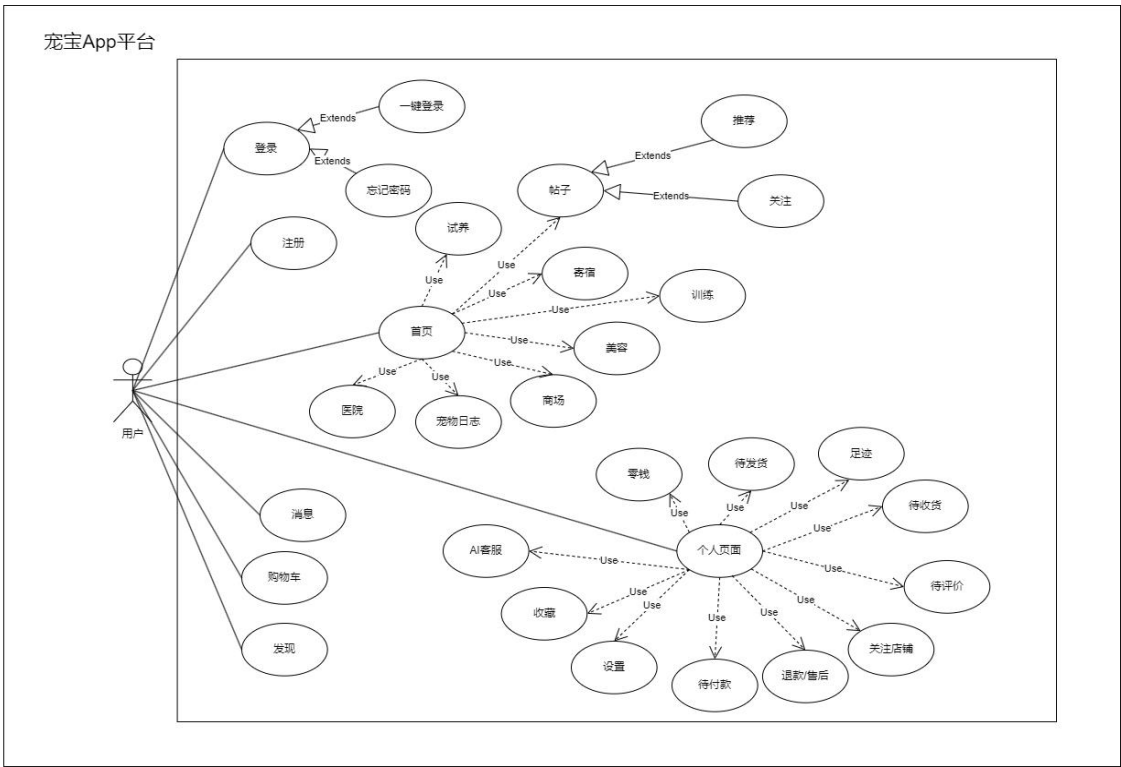


图 4.1 系统用例图

例有登录、注册、首页、个人页面、消息、购物车、发现等。其中扩展关系用例体现在了登录用例和帖子用例，登录包含了一键登录和忘记密码，帖子扩展了推荐和关注。而包含关系用例则在首页和个人页面可以看出。首页包含了试养、帖子、寄宿、训练、美容、商场、宠物日志和医院。个人页面包含了零钱、待发货、足迹、待收货、待评价、关注店铺、退款/售后、待付款、设置、收藏和 AI 客服。总之，透过用例图可以很好地观察并以很简单的方式简述系统功能。

4.2 功能模块划分及说明

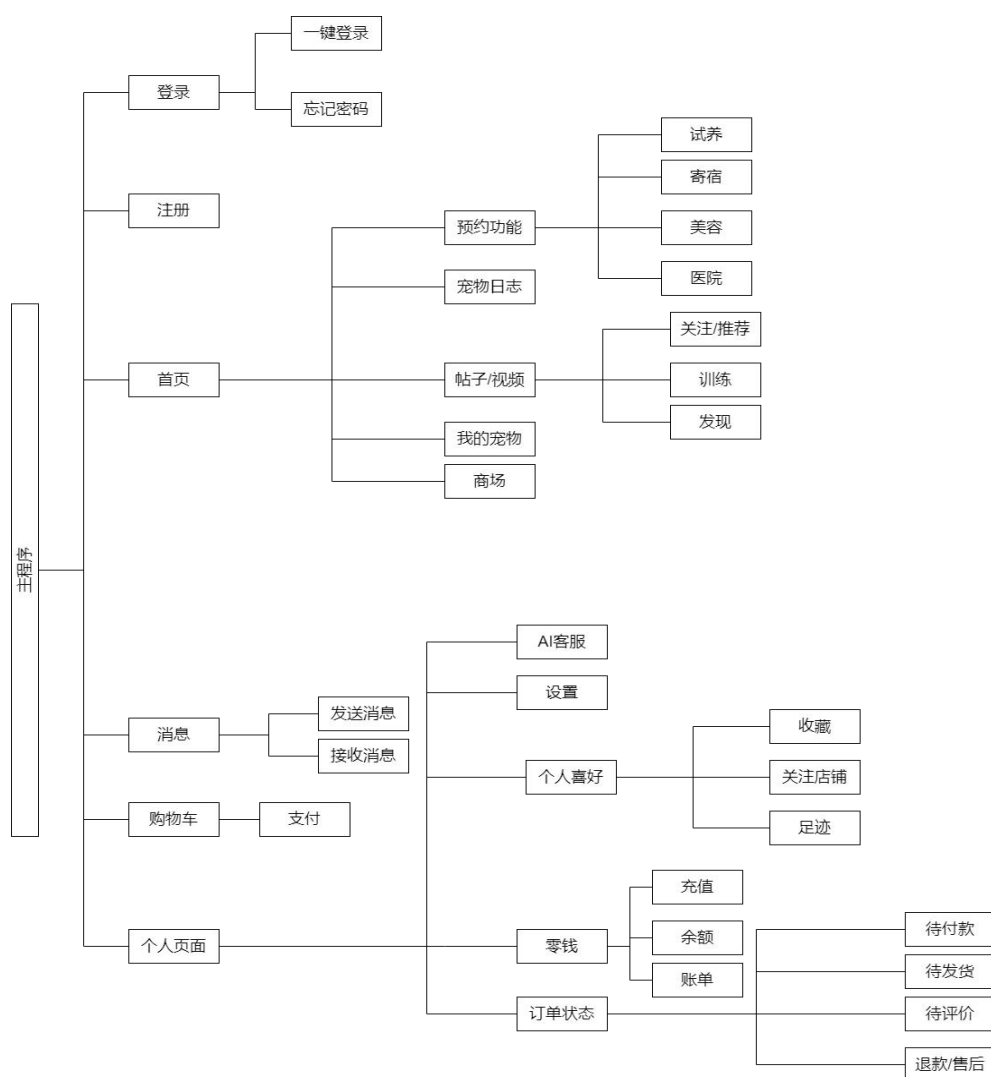


图 4.2 系统功能图

通过章节 4.2 的用例图看出了本文的软件平台应该如何设计，大致上应该具备了哪些功能。不难看出，本软件平台具备的功能繁多，因此不便于开发人员去读懂或者会遗漏了某些功能。因此本章节就是介绍本文的软件平台如何划分功能模块，便有了图 4.2 的系统功能图便于开发人员读懂本软件平台想要具备的功能，对接下来开发软件平台也有非常大的帮助。此外，一个有质量的程序必须具有适当的扇入扇出，模块之间的独立性要高，模块规模要适中等特点。因此，图 4.2 考虑了以上特点把本平台软件的功能进行划分。此功能图的深度为三层。

第一层的模块有登录、注册、首页、消息、购物车和个人页面。第二层的模块则是一键登录、忘记密码、预约功能、宠物日志、帖子/视频、我的宠物、商场、发送消息、接收消息、支付、零钱、订单状态。第三层的模块有试养、寄宿、美容、医院、关注/推荐、训练、发现、收藏、关注店铺、足迹、充值、余额、账单、待付款、待发货、待评价、退款/售后。登录模块是提供用户进行登录账号操作，从中，本文扩展了其功能可以让用户选择一键登录和忘记密码以方便用户再登录的过程中忘记密码或是更够更快速地登录。注册模块由于只需将用户输入的账户信息加入数据库，所以并没有下一层模块。

主页模块是非常主要的一个模块，其下一层模块有预约模块、宠物日志、帖子/视频、我的宠物和商场。其中预约功能模块和帖子/视频模块拥有第三层模块。其原因是试养、寄宿、美容、医院这几个模块都相同拥有预约的功能。关注/推荐、训练、发现也是同样原理，拥有播放帖子和刷视频的功能。消息模块的作用是给用户可以发消息和接收消息。购物车需要具备支付功能给用户进行付款。最后，个人页面可以提供给用户一些列个人账户管理的功能有 AI 客、设置、个人喜好、零钱、订单状态等状态。收藏、关注店铺、足迹等模块被归类为个人喜好模块的下一层关系^[13]。充值、余额、账单等模块是零钱的下一层模块。订单状态则是提供用户查看用户所购买的商品或服务状态，便有了待付款、待发货、待评价、退款/售后等下一层模块。

总而言之，本文是通过模块之间拥有相同的功能或者存在同一个页面以及结合了软件结构规则设计出了此功能图。

4.3 平台 ER 图以及数据库表设计

此小节是主要介绍了软件平台的数据库，数据库对于一个软件系统来说极为重要，负责数据的存储与管理、数据的检索与查询、数据的安全等。因此，设计一个数据模型图来描述本软件平台需要的数据库表，如图 4.3。



图 4.3 软件平台的整体数据库

图 4.3 是本软件平台《宠宝》的整体数据库图。如图所示, 包含了许多数据表和表与表之间存在这关系。从中, 数据表主要的数据表有 user, shops, follows, posts, message, chatRooms, payment, comment, items 和 cart 等非常多数据表。因此本文将对各表单独拆分出来介绍并以实体属性图表达和详细介绍。

首先, 表 4.1 是用户信息表, 是本软件平台最为重要的数据库表之一。分别记录着用户各类信息。本文的登录系统、注册系统、预约功能均需要此表来获取用户信息。在 Users 表里的属性包含了 Id, userName, userType, UserEmail, userPassword, userPhone, userAdress, userIdCard, userPhoto, pocketMoney。而 Id 作为本表的主键且不能为空以及自动增长。

表 4.1 Users 表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	用户 id
UserName	Varchar	255	Y		用户名
UserType	Varchar	255	Y		用户类型
UserEmail	Varchar	255	Y		用户邮箱
UserPassword	Varchar	255	Y		用户密码
UserPhone	Varchar	255	Y		用户电话
UserAddress	Varchar	255	Y		用户地址
UserIdCard	Varchar	255	Y		用户身份证件
UserPhoto	Text	-	Y		用户头像
PocketMoney	Int	11	Y		用户钱包

表 4.2 是商店的信息表, 储存着商家的基本信息、货品种类数量等。其中, 设有一个 UserId 的外键来表示商家本身的信息。本文用可以通过此表来获取并展示商店的名字、图片、详情等信息在前端。此表包含的属性有 Id, userID, shopName, shopType, shopAddress, shopContact, shopDetail, shopPhoto 和 shopItemQuantity。Id 为此表的主键且不为空以及自动增长。

表 4.2 Shops 表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	商家 Id
UserId	Int	11	Y	外键	用户 Id
ShopName	Varchar	255	Y		商家名称
ShopType	Varchar	255	Y		商家类型
ShopAdress	Varchar	255	Y		商家地址
ShopContact	Int	11	Y		商家联系方式
ShopDetail	Text	-	Y		商家详细情况
ShopPhoto	Text	-	Y		商家图片
ShopItemQuanty	Int	11	Y		产品/服务数量

表 4.3 的建立是为了记录用户购买商品或服务时的信息。通过此表可以得知用户想要购买的商品或者服务的信息、从哪个商家购买、购买数量等。在购物车页面需要调用这类信息，以使用户管理想要购买的商品及服务。此表包含的属性有 Id, ShopId,ItemId, UserId, Status, Quantity, Time, Location, IsPayment, ShopComment 和 ItemComment。Id 为此表的主键且不为空以及自动增长。

表 4.3 Cart 购物车信息表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	购物 Id
ShopId	Int	11	Y	外键	商家 Id
ItemId	Int	11	Y	外键	商品/服务 Id
UserId	Int	11	Y	外键	用户 Id
Status	Varchar	255	Y		订单状态

表 4.3 (续)

字段名称	数据类型	长度	是否为空	键	备注
Quantity	Int	11	Y		数量
Time	Timestamp	-	Y		服务的预约时间
Location	Text	-	Y		服务的地点
IsPayment	Boolean	2	Y		付款
ShopComment	Text	-	Y		评价商家
ItemComment	Text	-	Y		评价商品/服务

表 4.4 记录着用户的付款信息。此表作用可以显示哪个用户还未付款成功 and 哪位用户付款成功了。此表的属性包含了 Id, UserId, CardId, CardNo, CardHolder 和 Cv1。Id 为此表的主键且不为空以及自动增长。UserId 作为外键表示哪个用户进行了付款并是否付款成功。

表 4.4 Payment 付款表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	付款 Id
UserId	Int	11	Y	外键	用户 Id
CardId	Int	11	Y		银行卡 Id
CardNo	Int	11	Y		银行卡号码
CardHolder	Varchar	255	Y		银行卡持有人
CV1	Varchar	255	Y		银行卡安全码

表 4.5 是为了储存用户发的帖子信息，其作用是记录每一条用户所发的帖子。此表的属性包含了 Id, Title, Body1, Body2, UserId, PostTime。UserId 作为外键是为了区分是哪个用户发了贴。Id 为此表的主键且不为空以及自动增长。

表 4.5 Posts 帖子信息表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	分享 id
Title	Varchar	11	Y		标题
Body1	Text	-	Y		内容 1
Body2	Text	-	Y		内容 2
UserId	Int	11	Y	外键	用户 id
PostTime	Timestamp	-	Y		时间

表 4.6 的作用是记录每条的消息的信息和内容，SenderId 是表示哪个用户发送的消息。ChatId 这是指发送到哪个聊天室，因此此表与图 4.10，表 4.7 ChatRooms 数据表有着极为重要的关系，缺其一就会导致本软件平台的消息功能失效。此表的属性包含了 Id，senderId，chatId，sendTime，Body 和 Issent。Id 为此表的主键且不为空以及自动增长。

表 4.6 Message 消息表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	消息 Id
SenderId	Varchar	255	Y		发送者 Id（用户）
ChatId	Int	11	Y	外键	聊天 Id
SendTime	Time		Y		发送时间
Body	Text		Y		内容
Issent	Bollean	2	Y		是否发送成功

就如上一段所述，聊天室表即表 4.7 的作用是集合了每个发来的消息的信息的一个集合。就是利用 messageId 为外键获取对应的消息信息。因此图 4.10

和表 4.7 需要包含的属性有 Id, UserId, chatName 和 IsSeen 以及 Id 为本表的主键且不为空以及自动增长。

表 4.7 ChatRooms 聊天室表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	
UserId	Int	11	Y	外键	
ChatName	Int	11	Y		
isSeen	Boolean	2	Y		是否已读
messageId	int	255	N	外键	

表 4.8 是用来记录每一个商品或服务的信息,商家表很需要通过此表来获取对应的货品或服务信息来显示此店家有什么商品以及其数量。本文对此表设计的属性有 Id, shopId, ItemType, ItemName, ItemStock, ItemPrice, ItemPhoto 和 ItemDetail。其中 Id 为此表的主键且不为空以及自动增长。

表 4.8 Item 商品服务表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	商品/服务 Id
ShopId	Int	11	Y	外键	商家 Id
ItemType	Varchar	255	Y		商品/服务类型
ItemName	Varchar	255	Y		名称
ItemStock	Int	11	Y		商品库存
ItemPrice	Decimal	10	Y		价格
ItemPhoto	Text		Y		照片
ItemDetail	Text		Y		内容

表 4.9 用来记录用户关注和收藏了哪些店、铺帖子和用户。其包含的属性有 UserId, FollowingUserId, FollowedShopId, SavedPostId 和 SavedItemId。其中, UserId 是表示着用户, FollowingUserId 是表示着正在关注的其他用户,

FollowedShopId 是指用户关注的店家，SavedPostId 为用户所收藏的帖子以及 SavedItemId 是用户收藏的服务或商品。

表 4.9 Follows 关注收藏表

字段名称	数据类型	长度	是否为空	键	备注
Following UserId	Int	11	Y	外键	关注的其他用户 Id
UserId	Int	11	Y	外键	用户 Id
Followed ShopId				外键	关注的商家 Id
Saved PostId				外键	收藏帖子 id
Saved ItemId				外键	收藏产品/服务 id

表 4.10 是用来存储用户所发表对商家和货品的评论。此表包含的属性有 Id,UserId,ShopId 和 body。UserId 是指用户本身，ShopId 为用户要评价的店家，以及 Body 为评论的内容，其可以是文本或者照片等。

表 4.10 Comment 评论表

字段名称	数据类型	长度	是否为空	键	备注
Id	Int	11	N	主键	
UserId	Int	11	Y	外键	
ShopId	Int	11	Y	外键	
Body	Text		Y		是否已读
字段名称	数据类型	长度	是否为空	键	备注

以上介绍了本软件平台的所有的数据表，但是还未体现出数据表与数据表之间的联系。因此接下来，本文设计了本软件平台的整体 ER 实体关系图表述数据表与数据表之间的关系。如图所示，本软件平台是以 Users 为中心。以下表 4.11 为数据表与数据表即实体之间的关系：

表 4.11 实体与实体的关联

实体 1	实体 2	关联	关系
User 表	Follows 表	关注	n: n
User 表	Posts 表	创建	1: n
User 表	Payment 表	付款	1: n
User 表	ChatRooms 表	拥有	n: n
User 表	Shops 表	持有	n: n
User 表	Comment 表	评论	n: n
User 表	Cart 表	预定商品和服务	1: n
ChatRooms 表	Message 表	发送/接受	n: n
shops 表	Items 表	销售	1: n

以下图是根据表 4.11 划分并设计出的 ER 数据库关系实体图。分为几个主要的部分图 4.4 为用户与商家所需的后端 ER 关系图，图 4.5 是用户购买本软件平台的商品或服务的相关 ER 实体关系图，而图 4.5 是本软件平台提供用户与社区交流以及通讯所需的 ER 实体关系图。

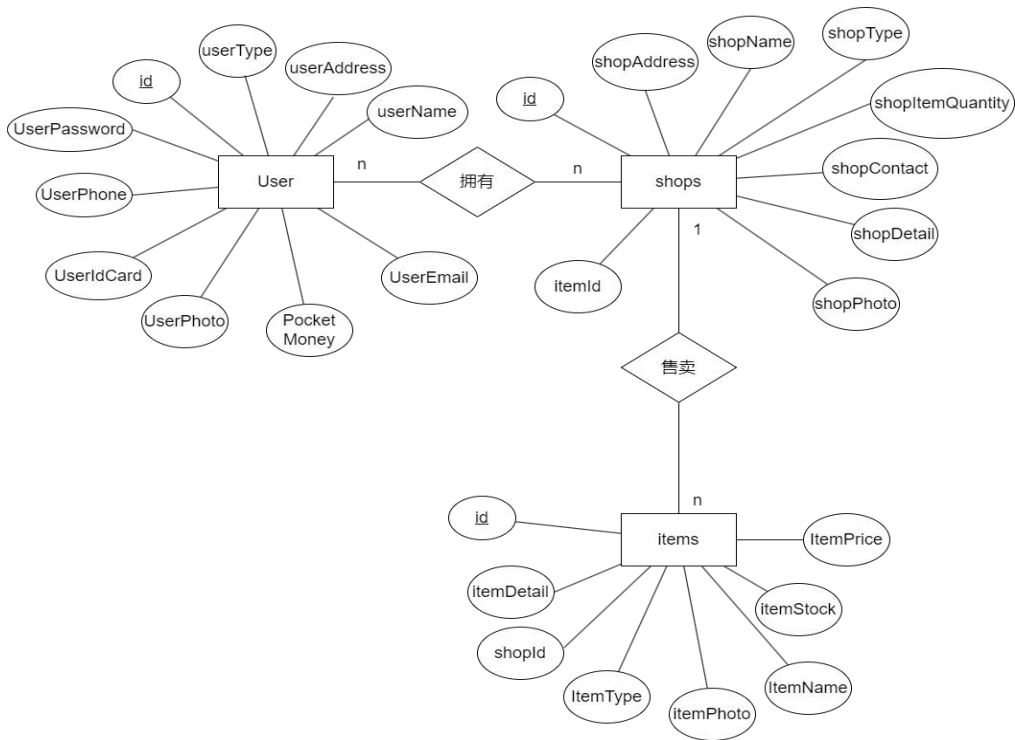


图 4.4 为用户与商家 ER 图

如图 4.4 所示，User 表 和 Shops 表是多对多的关系，Shops 表与 Items 表是一对多的关系。这是表示这一个用户可以拥有很多商店，而商店可以售卖很多的商品。这里，本文使用的 User 表里的 userType 来区分用户的类型即是店家或是顾客。

而图 4.5 是用户在使用本软件平台购物时需的数据库 ER 实体关系图。如图 4.5 所示 User 表与 Cart 表是一对多的关系，User 表与 Comment 表是多对多的关系，User 表与 Payment 表是一对多的关系。这表明一个用户可以加入很多商品到很多购物车里，本文是以 cart 里的 Itemid 属性作为外键来绑定哪个商品被加入了进了购物车。接着是很多用户可以发布很多评论，评论也可以被很多用户发布。而一个用户能拥有很多付款单 Payment 而 payment 只能一个用户付款。

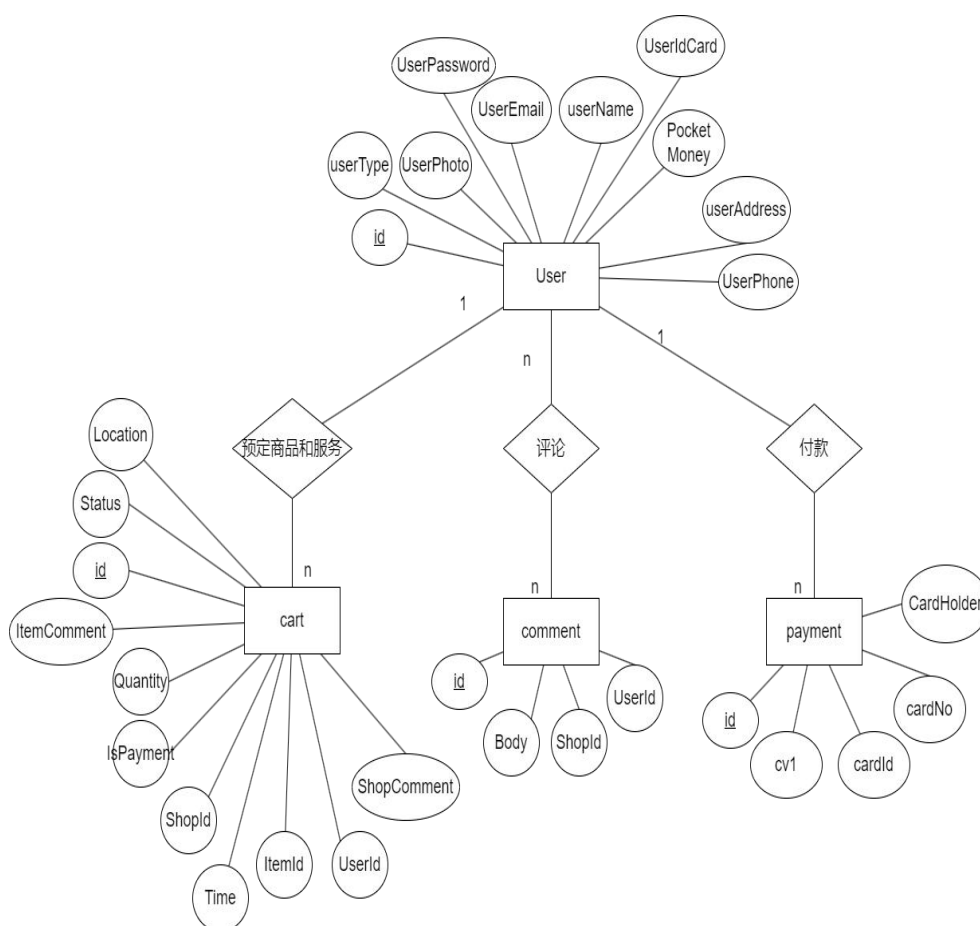


图 4.5 用户购物 ER 实体关系图

最后是图 4.6，为实现用户通讯和社区交流所设计的 ER 实体关系图。在图里 User 表与 follows 表是多对多关系，User 表与 Posts 表是一对多关系，User 表与 ChatRooms 表是多对多关系，ChatRooms 表与 Message 表是多对多关系。其中表示着用户可以关注很多用户、商店和帖子，而用户、商店和帖子也可以被很多用户关注收藏。接着是一个用户可以创建很多帖子，而一个帖子只能一个账号发帖。再来用户拥有很多聊天室，聊天室也可以拥有很多用户。补充的还有聊天室可以发送很多信息，而信息可以被很多聊天室发送或接受。因此用户是通过查找想要的接收者的聊天室并对其进行发送或接受消息。

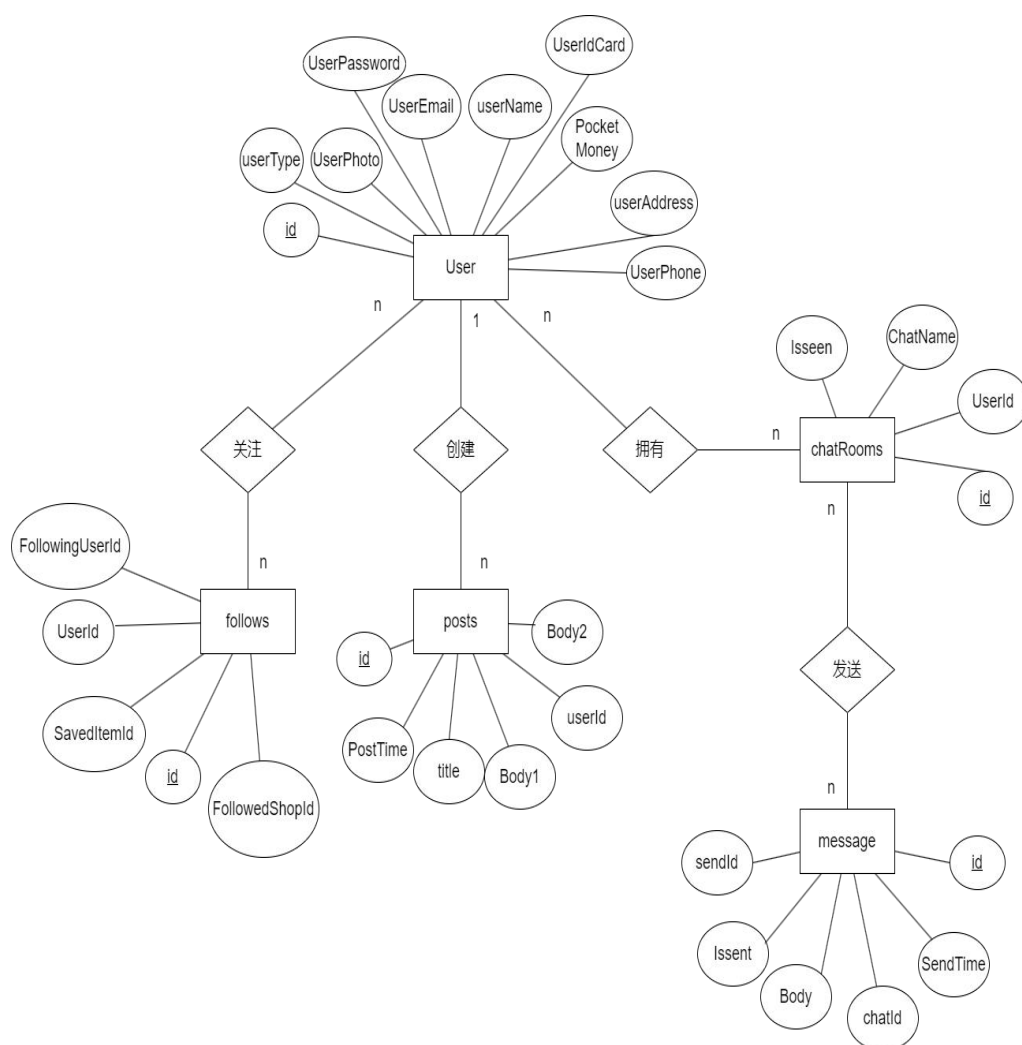


图 4.6 用户通讯和社区交流功能 ER 实体关系图

4.4 本章小结

本章节介绍了“如何实现软件平台”的概念。首先对需求分析分析出了用户的需求功能，设计出了软件平台的系统中架构即系统用例图，再对其进行分析研究，划分了其功能模块并设计出了功能图。之后本章节还设计出了软件平台所需的数据和之间的关系。

第五章 功能开发

前一个章节已介绍了该如何开发本软件平台，接下来本章节将讲解如何具体实现本软件平台。本章节会使用时序图、顺序图、图表与部分主要的伪代码来展示如何实现本软件平台。流程图可以很好地展示本软件平台的在运行某一个功能或界面时的操作步骤和流程。而时序图是用来展示对象之间的交互及其消息传递顺序和参与者与本软件平台的交互。

5.1 各界面时序图与流程图

通过上一章的功能模块分析以及划分后，本文将以其为基础，设计了用户操作该功能模块时的整个流程。本文将以流程图和时序图来展示。流程图用于描述一个过程或系统中的各个步骤、活动、决策或事件之间的关系和顺序，常用于分析和设计系统、工作流程、算法等，以便更好地理解 and 优化过程。时序图则主要用于展示系统中各个对象之间的交互和消息传递顺序，通常用于描述软件系统、通信系统等中的对象之间的时序关系。

图 5.1 展示了本软件平台的注册和登录的功能流程图。首先本软件平台会默认将用户带到登录揭秘那，用户会被要求输入密码和账户。平台将会判断用户输入的账户是否存在，如果该用户未曾注册本平台，平台会将此时的界面跳转到注册功能界面并要求用户进行注册账户。用户需要根据本软件平台的要求和格式输入个人一些基本信息等。然后平台会对所输入的信息进行核对等，如果失败平台会要求用户重新输入直至用户达到格式和标准的要求。接着，只有在用户注册成功时才能到登录功能进行登录操作，否则用户将会一直被判断成注册失败并要求用户再注册一次。当用户注册成功后，平台会将用户信息进行保存，并把用户带到登录界面，此时用户再次输入所注册的用户信息时，平台才会判断用户确实存在的，平台将会判断密码是否正确，如果用户的密码与注册时的密码是正确就登录成功进入代主页，否则登录失败并要求用户在输入一次。

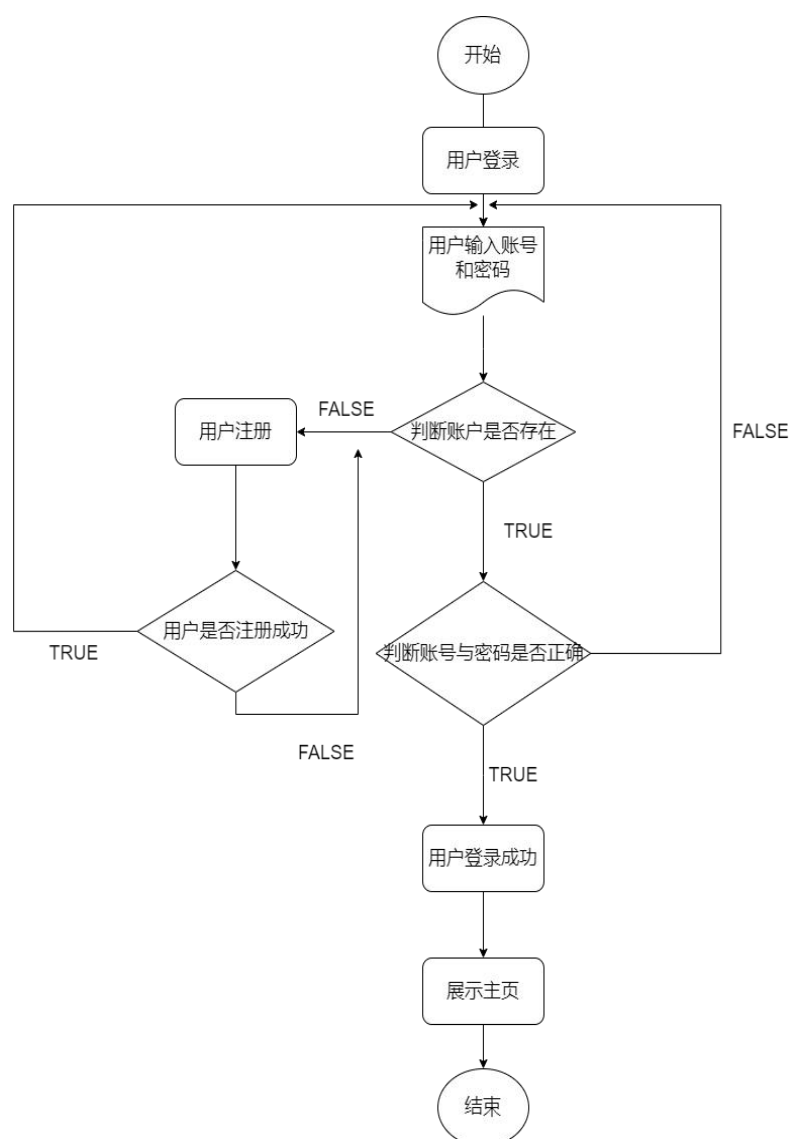


图 5.1 用户登录与注册流程图

图 5.2 是用户登录时序图。用户首先会需要输入用户名和密码，如何平台会将用户名和密码传给登录系统。接着，登陆系统会接收用户和密码并向数据库进行查询，如果用户名或密码不正确将会返回错误信息给用户。用户需要重新输入，然后平台，登录系统和数据都会执行同样操作直至用户输入正确的用户名和密码，才会返回成功登录给用户。

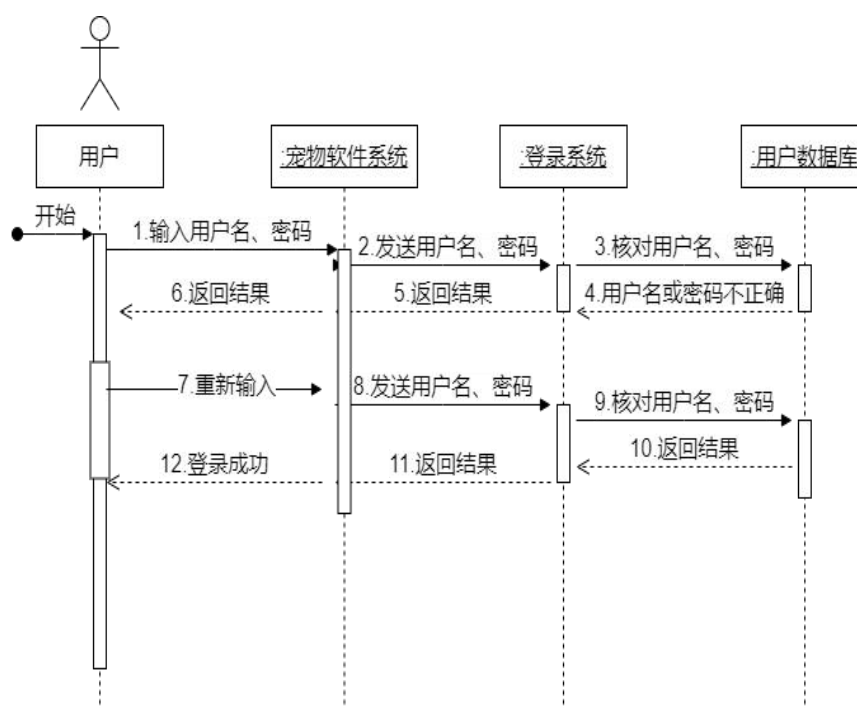


图 5.2 用户登录时序图

图 5.3 所展示的是用户在进行注册时的时序图。首先用户输入用户信息到平台，平台传送信息给注册系统，注册系统将会核对用户所输入的信息格式，如果不正确将会返回注册失败的结果给用户，此过程会一直重复直至用户输入了正确格式的信息，注册系统才会把用户信息传入数据库并更新用户数据库。更新完毕后会返回注册成功结果给用户。

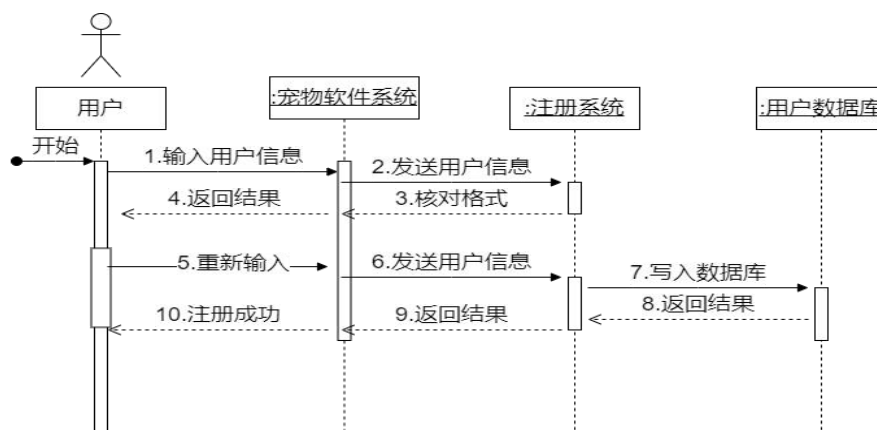


图 5.3 用户注册时序图

图 5.4 是本平台的试养宠物的流程图。用户一开始进入试养界面便可以选着想要并感兴趣的宠物进行一段时间的试养。在选着完试养的宠物后，平台会对用户进行实名认证的检查，查看用户是否已实名认证，没有则用户被要求填写身份证，用户如果没有进行此操作便不能试养该宠物也不能进行下一步的操作。如此，能大大避免了用户会虐待所试养的宠物和一些法律纠纷也是本软件平台与用户之间能有更可靠的信用。在平台确定用户通过了实名认证之后便选择试养时段并提交试养申请到平台，平台就会处理通知商家，计算费用等。用户在缴纳完费用后便可以指定的地点领取宠物并开始试养宠物。

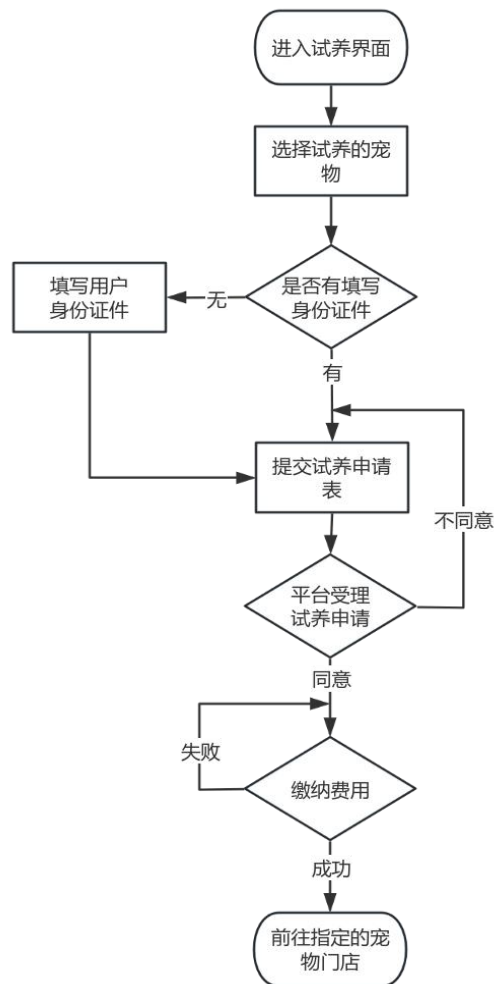


图 5.4 试养宠物流程图

图 5.5 是试养宠物的时序图。首先用户通过平台发送请求进入试养界面，平台会向试养界面发送请求，请求到达后，平台会返回试养界面给用户。此时，用户会看到试养界面所提供的功能如提供很多商店和各类的试养宠物提供给用户选择。接着，用户会选定感兴趣的宠物，查看该宠物的详细信息，以及该宠物的所在地和其商店。在用户选择并确认想要试养该宠物并且已平台指定实名认证后，平台就会发送通知给相关的店家。商家在收到通知后可以选择是否接受交易。如接受，平台会通知用户，用户就能选择试养时段。用户在确认时段后，试养界面要求更新试养宠物在数据库的信息，将宠物的信息改为已被饲养，并更新了试养界面，不会再有该宠物提供给用户选着。最后平台会返回成功界面给用户提示用户该宠物的所在地，用户就可以到线下领取试养宠物或者选着邮寄的方式等寄到用户手里。

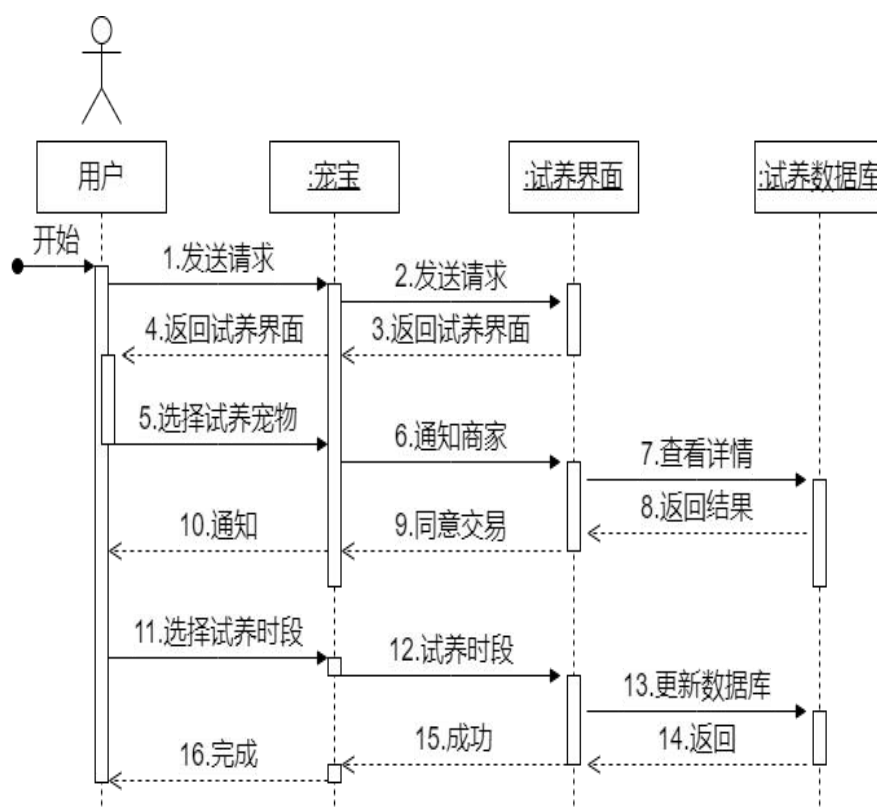


图 5.5 试养宠物时序图

图 5.6 分别对应平台的寄宿、美容和医院的流程图。这些流程图极为相似，因为他们的主要进行预约功能。首先，用户进入对应的界面，平台会推荐一些列的商家、商品和其对应页面的服务，如寄宿就是提供各类的寄宿套餐和寄宿商家，医院界面就推荐一系列有合格准证并且所在地靠近用户的医院，美容界面则是一些列美容商家。如此用户就可以选择想要的商家或者想要的服务。之后，用户就可以进行预约。在预约时想要用户填写预约表，即选定预约时间，预约的服务或套餐。在之后，平台会自动进行费用的计算，用户此时还是能够选择是否继续预约。在确认预约并成功后用户就需支付刚刚平台提示的费用，支付成功后就可以前往线下对应的商店。

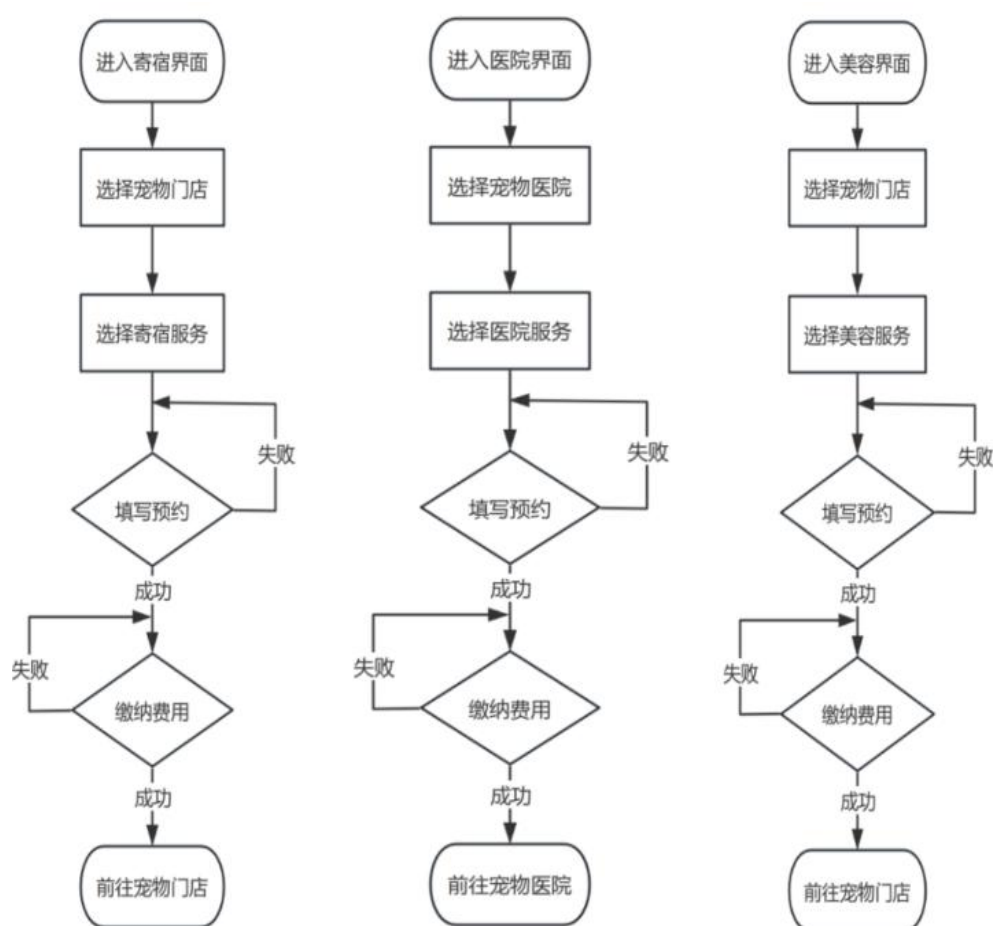


图 5.6 宠物寄宿、宠物医院、宠物美容流程图

图 5.7、图 5.8、图 5.9 也是分别对应平台的寄宿、美容和医院的时序图。同理，这些时序图都非常相似。首先，用户通过平台发送进入对应页面的请求。平台接受到请求后便会向对应的界面发送请求，再来平台就会返回对应界面给用户。接着，对应的界面就会展示对应的商品、服务和店家提供给用户选择。如果用户选择或查询商家详细信息时，平台便会通过对应的界面向发送数据库获取对应商家信息的请求。在平台成功获取信息成功后便会返回给用户。接着用户便可以选择其商家信息里的服务，如医院就是就诊，美容就是洗澡，寄宿就是宠物住宿。之后，用户申请预约，用户会填写预约申请表信息，在平台收到后，平台便会通知商家。如商家确定接受该预约后，平台会更新数据库，像是该店家已是否爆满等。最后，平台就会通知用户预约完成并且平台会计算出所需的费用。之后用户进行支付或者到现场支付，也可以取消预约。最后平台通知商家并更新数据库如用户是否与付款。用户是否已取消预约等。整个流程便完毕。

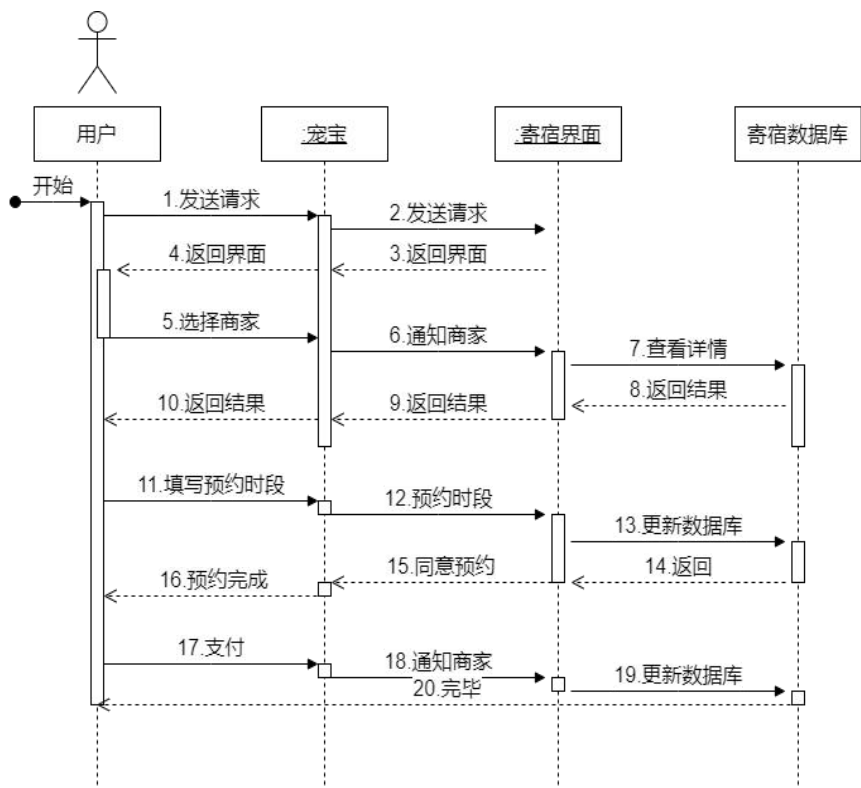


图 5.7 宠物寄宿时序图

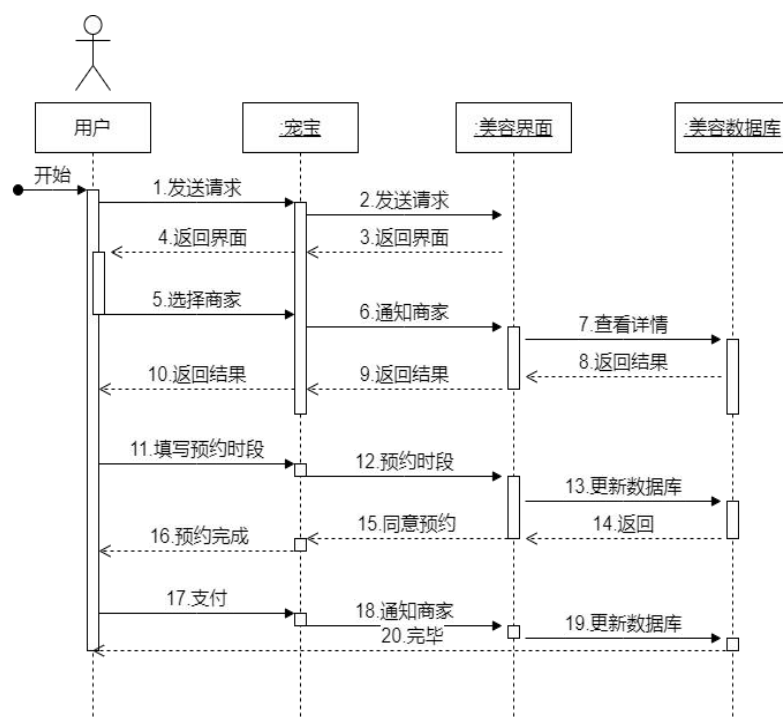


图 5.8 宠物美容时序图

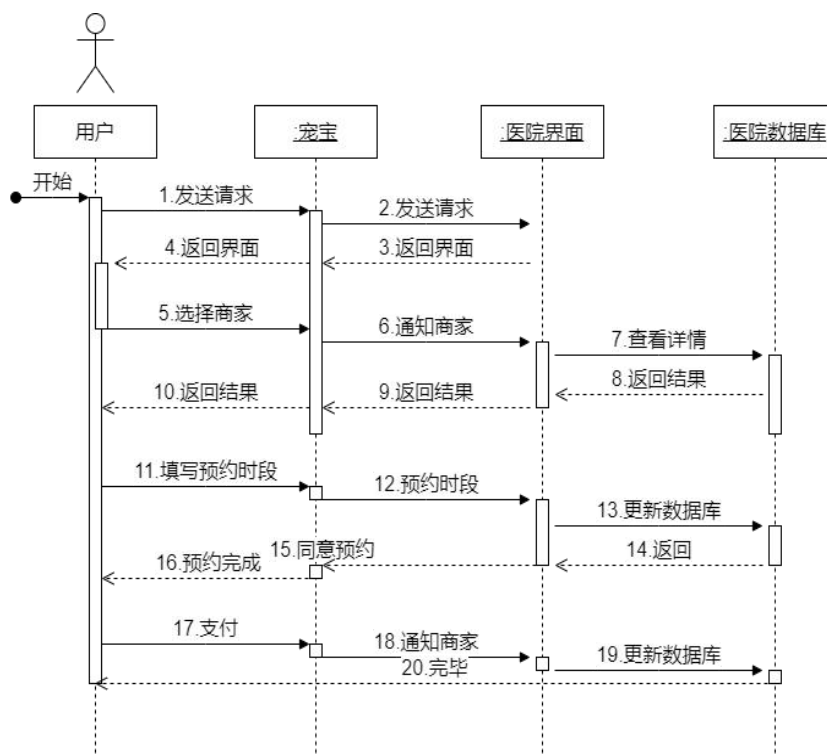


图 5.9 宠物医院时序图

图 5.10 是本平台的商城流程图，首先，用户进入商场页面，选择想要购物的商品，便支付费用，成功就等待发货，失败这会被要求支付费用。

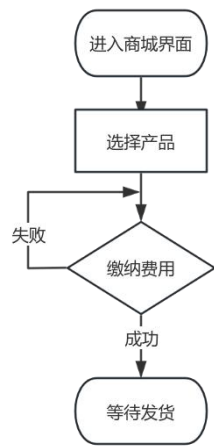


图 5.10 商城流程图

图 5.11 展示了商场的时序图，用户发送进入商场页面请求，平台向商店系统获取商场页面结果后返回给用户。接着，用户选着商品并发送购买请求给平台，平台通知商家后商家查询货物数据库，如果货物充足，商家同意交易，平台返回同意交易结果给用户，用户便进行支付操作，平台负责传送支付信息给商家，商家更新货物数据库，完毕后平台返回支付成功给用户。

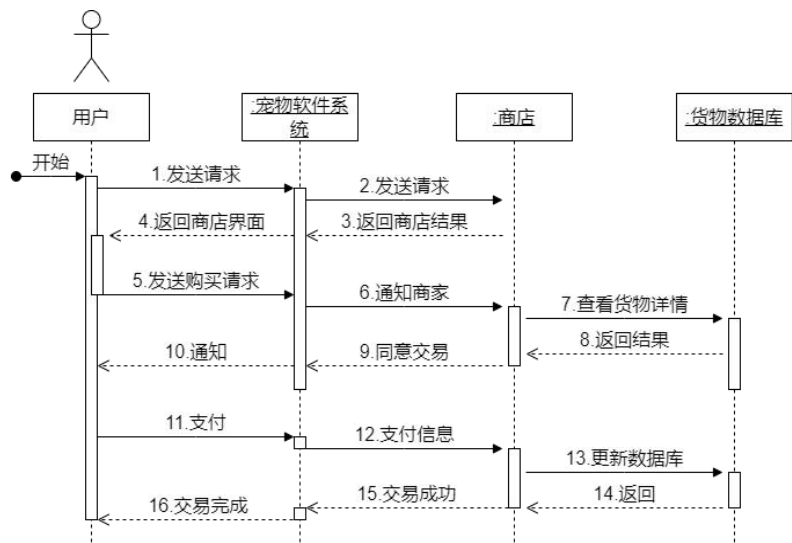


图 5.11 商场时序图

5.2 前端实现与开发

图 5.12 是本软件平台的总入口，也是唯一入口。如此能提高程序的维护性、安全性和扩展性。代码中设定了整个平台的主题和背景颜色。然后并把主页面设为登录页面。所以，程序会优先进入登录页面就在图 5.12 里的第 19 行代码 `home: const Login()`。其中 `const` 就是固定只运行 `Login()` 函数即登录界面，不会改变。第一行至第三行则是导入了必要的包，如 `themes.dart` 是用来设置本软件平台的主题颜色的包，其也能设置在移动端开启夜间模式下，本软件平台的主题界面应该如何呈现。而第二个包是 Flutter 框架编写时必要导入的包，其包含了全部 flutter 框架的全部功能函数。接着 `main()` 就是主函数，其运行了 `MyApp()` 函数。`MyApps` 在第九行代码就是继承了 Flutter 的 `StatelessWidget` UI 组件。在第 9 行代码后就是对其界面的一个设置。

```

1  import 'package:app/shared/resources/themes.dart';
2  import 'package:flutter/material.dart';
3  import 'screens/login_page/login.dart';
4
5  void main() {
6    runApp(const MyApp());
7  }
8
9  class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11
12   // This widget is the root of your application.
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(
16       debugShowCheckedModeBanner: false,
17       title: 'Test',
18       theme: AppThemes.LightTheme(),
19       home: const Login(),
20     ); // MaterialApp
21   }
22 }
23

```

图 5.12 本软件平台总入口

表 5.1 是实现了登录界面的伪代码，源代码过长且较难读懂表示，所以本文便以伪代码的方式呈现撰写源代码的过程，并介绍几个主要伪代码。伪代码中的 `StatefulWidget` 是 Flutter 框架中的一个重要概念，用于构建具有交互性的用户界面。在 Flutter 中，界面的状态可以分为两种：静态状态（`Stateless`）和动态状态（`Stateful`）。接着，就是使用 `Build()` 函数进行了定义页面布局。其中在这个登录界面设置了用户名输入框、密码输入框、忘记密码和注册账户的链接和设置了对应按钮监听事件。当用户输入账号或密码后，点击对应按钮就会触发事件监听。平台就是进行链接数据库来核对用户输入的密码和账号是否存在和正确，如果正确这是带领用户跳转到主页。用户点击的忘记密码、一键登录或者注册账户都是使用到 `ontap()` 方法来实现界面的跳转和设置想要执行的任务

表 5.1 伪代码-登录界面

伪代码 - 登录页面

1. 导入必要的包和文件
2. 定义 `LoginBody` 类，即登录页面的内容，继承自 `StatefulWidget`;
3. 创建全局表单键;
4. 定义布局页面的方法 `build()`;
5. 编写输入框、按钮连接等小部件 `Form()`，包含各种子部件
6. 用户名输入框
7. 密码输入框
8. 忘记密码和注册链接
9. 登录按钮
10. 一键登录按钮
11. 调整布局以及美化布局
12. 添加按钮监听事件 `onTap()` 并设置事件。

表 5.2 是主页面的伪代码，伪代码中 HomePage 类为主页的主函数，build 类为画布，可以用来装各种小部件如 Scaffold, SingleChildScrollView ,container 等。主页底部设有自定义设计的底部导航 BottomNavigationWidget。BottomNavigationWidget 为外部类，且常常被调用在其他页面中。因此本文为了统一本软件平台的设计风格制定编写了一个 BottomNavigationWidget 并在需要的时候调用它，减少了许多代码量。而 Scaffold, SingleChildScrollView , container 都是 Flutter 自带的函数和方法，需要根据其语法进行编写。SingleChildScrollView 是可以使界面在一个方向（垂直或水平）滚动一个方法。

表 5.2 伪代码 -主页界面

伪代码-主页界面
<ol style="list-style-type: none"> 1. 导入必要的包和文件 2. 定义 HomePage 类，继承自 StatefulWidget 3. 定义状态对象 4. 定义 initState 方法 5. 监听页面变化 6. 定义 dispose 方法 7. 释放资源 8. 定义 build 方法 9. 返回 Scaffold 小部件，包含主体和底部导航栏 10. 主体为 SingleChildScrollView 包裹的 Container，包含各种子部件 11. 地标和扫描图标 12. 搜索框 13. 喇叭通知今日热点 14. 标志性图标网格 15. 轮播图 16. 四宫格卡片 17. 底部导航栏为 BottomNavigationWidget 小部件

表 5.3 是实现试养页面的伪代码。代码中 AppBar 为用于在应用程序顶部显示标题、操作按钮和其他相关内容。在代码中，AppBar 用来包装返回按钮、标题和搜索栏。CustomWidget 是自定义的一个小部件，其在代码里的作用是用于播放视频。ListView 是一个 flutter 自带部件是一个卡片列表。

表 5.3 伪代码-试养页面

伪代码-试养页面
<ol style="list-style-type: none">1. 定义一个 Stateful 组件 TryCardBody2. 创建全局表单键3. 定义 build 方法4. 返回 Scaffold 小部件，包含 AppBar、背景颜色和主体部分5. AppBar 部分包含返回按钮、标题和搜索栏6. 主体部分为 Column，包含轮播图、按钮列表和卡片列表7. 轮播图部分为 CustomSwiper 小部件，展示轮播图8. 按钮列表部分为 Row，包含两个 Flexible 容器9. 第一个 Flexible 容器包含 ListView.builder，展示按钮列表10. 第二个 Flexible 容器包含 Column，展示卡片列表11. 卡片列表部分为 ListView.builder，展示视频卡片列表12. 每个视频卡片包含视频缩略图、播放按钮和视频信息13. 定义 _buildButton 方法14. 返回一个 GestureDetector 小部件，包含文本按钮15. 点击时触发 _onButtonPressed 方法16. 定义 VideoWidget 类，继承自 StatefulWidget17. 定义视频播放相关的状态和控制器18. 定义 initState 方法19. 初始化视频播放控制器20. 定义 dispose 方法21. 释放视频播放控制器22. 定义 build 方法23. 返回一个 Column 小部件，包含视频缩略图、播放按钮和视频信息

5.3 后端数据库与接口

表 5.4 是创建数据的各表，由于本软件平台的后端开发步骤非常庞大，本文无法完整呈现，所以以下伪代码是实现本软件平台的后端部分步骤且较为重要。在连接数据库并创建数据表后，PrimaryKey 是将其属性设为主键，ADD FOREIGN KEY 是讲其设为外键，关系数据库做准备。

表 5.4 伪代码-创建数据库表

伪代码-创建数据库表:

1. 连接数据库服务器
2. 创建数据库如果不存在，CREATE DATABASE IF NOT EXISTS
3. 选择数据库 '示例数据库 USE “ ”
4. 创建各类表表-例：CREATE TABLE `users`
(`id` integer PRIMARY KEY,
`username` varchar(255),
`userType` varchar(255),
`userEmail` varchar(255),
`userPassword` varchar(255),
`userPhone` varchar(255),
`userAddress` varchar(255),
`identityCard` varchar(255),
`Userphoto` varchar(255),
`pocketMoney` integer),
5. 建立或更新表之间的关联-例：ALTER TABLE `users` ADD FOREIGN KEY
(`id`) REFERENCES `cart` (`userId`);
6. 插入数据-Insert into *
7. 查询数据-Select from *
8. 删除数据-Delete from *

表 5.5 是连接数据库的关键代码。其中 `updateUser` 是一个在修改数据库里的 `User` 表里的数据。代码里还设置了一个响应代码来检查数据库是否连接成功 200 为成功。其他需要对数据库进行操作的都需要用到此代码。

表 5.5 伪代码-连接数据库操作

伪代码-连接数据库操作
<ol style="list-style-type: none">1. // 定义一个函数，更新用户信息并返回更新后的用户模型，(<code>UserModel user, BuildContext context</code>):2. // 定义请求的 URL-Url = "<code>http://localhost:8080/updateuser</code>"3. // 发送 PUT 请求到指定的 URL，并传递用户信息作为请求体-定义 变量 <code>response</code> = 发送 HTTP PUT 请求 到 Url, 使用以下头信息和请求体:4. 头信息:<code>Content-Type: application/json</code>5. 请求体:将 <code>user</code> 转换为 JSON 字符串6. // 获取响应的字符串形式: 定义 变量 <code>responseString = response.body</code>7. // 如果响应状态码为 200，表示请求成功8. // 显示一个对话框，展示后端返回的响应信息9. 函数 创建对话框(<code>dialogContext</code>):10. 返回 一个警告对话框, 显示标题 '<code>Backend Response</code>' 和 内容 <code>response.body</code>

表 5.6 是在 Spring Boot 里构建 RestFul Apls 来进行数据库的访问和 CRUD（增、删、改、查）。`Service` 类用于处理业务逻辑，协调多个 `Repository`，并对外提供服务。`Interface` 类用于与数据库进行交互。`Model` 类用于表示应用程序中的数据对象或实体，通常与数据库表或其他数据源中的数据结构对应。`Controller` 类用于处理外部请求，调用 `Service` 层方法，并返回响应结果。这些只是部分代码。

表 5.6 创建 SpringBoostAPIs

伪代码: 创建 SpringBoostAPIs

1. 定义一个用户服务类: UserService: userRepository = 用户仓库
2. 定义一个函数, 用于注册用户: 注册用户(User user):
 返回 userRepository.保存(user)
3. 定义用户仓库接口, 继承自通用的 JPA 仓库接口 UserRepository
4. 定义一个用户实体类 User:
 主键 变量 id 类型 Long, 生成策略 = IDENTITY
5. 定义用户名字段
6. 定义 getter 和 setter 方法
7. 函数 获取用户名(): 返回 username
8. 函数 设置用户名(String username): 赋值 this.username = username
9. 定义一个用户控制器类 UserController:
10. 自动装配 变量 userService = 用户服务
11. POST 请求 映射 到 "/register"
12. 函数 注册用户(请求体 User user):
13. 变量 registeredUser = userService.注册用户(user)
14. // 返回包含已注册用户信息和 HTTP 状态码 201 的响应
15. 返回 新的响应实体(registeredUser, HTTP 状态.已创建)

5.4 本章小结

本章节讲述了具体实现软件平台的过程。页面设计部分, 本文用了时序图和流程图来展示用户的操作操作过程, 数据的流动。并给出几个核心界面的代码和伪代码为例子。后端部分也是如此。

第六章 运行与测试

经过了前几章的铺垫，本章就是来展示实践运行成果。以实际案例验证本软件平台的功能性、可用性和实用性。

6.1 功能与性能测试

本小结将对本文的软件平台进行一些有效的测试，以确保软件平台的质量、可用性、和性能。本文将采用功能测试和黑盒测试的测试方法这两方面来测试本软件平台。以下表 6.1 为本文的测试用例表：

表 6.1 测试用例表

测试用例 ID	测试项	采用的测试方法		与预期是否一致		
		输入	预期输出	运行结果		
TC001	用户注册	功能测试	有效的用户名、密码和邮箱	提示注册成功，跳转至登录页面	注册成功，跳转至登录页面	是
TC002	用户登录	功能测试	已注册的用户名和密码	提示登录成功，跳转至应用主页	登录成功，跳转至应用主页	是
TC003	发现功能	黑盒测试	点击“发现”按钮	进入发现页面，显示宠物相关内容	进入发现页面，显示宠物相关内容	是
TC004	试养功能	黑盒测试	点击“试养”按钮	进入试养页面，显示试养服务商家信息	进入试养页面，显示试养服务商家信息	是

表 6.1 (续)

测试用例 ID	测试项	采用的测试方法		与预期是否一致		
		输入	预期输出	运行结果		
TC005	寄宿功能	黑盒测试	点击“寄宿”按钮	进入寄宿页面，显示寄宿服务商家信息	进入寄宿页面，显示寄宿服务商家信息	是
TC006	美容功能	黑盒测试	点击“美容”按钮	进入美容页面，显示美容服务商家信息	进入美容页面，显示美容服务商家信息	是
TC007	医院功能	黑盒测试	点击“医院”按钮	进入医院页面，显示医疗机构信息	进入医院页面，显示医疗机构信息	是
TC008	商城功能	黑盒测试	点击“商城”按钮	进入商城页面，显示宠物用品	进入商城页面，显示宠物用品	是
TC009	客服支持	黑盒测试	点击“客服”按钮	进入客服支持页面，显示在线客服列表	进入客服支持页面，显示在线客服列表	是

除了黑盒测试与白盒测试外，本文也对本软件平台进行了性能测试并记录在表 6.2 中。

表 6.2 性能测试表

技术指标	评估方法	评估结果
运行速度	页面加载速度、功能响应时间	页面加载速度快，功能响应迅速。
安全性	数据加密、权限管理	用户数据得到有效加密保护，权限管理严格。
扩展性	新增功能、用户请求处理	可轻松新增功能，能够处理大量用户请求。
部署方便性	部署流程简单、快速	部署流程简单，仅需几步即可完成。
可用性	界面友好、操作简单	用户界面友好，操作简单易懂。

6.2 成果展示

登录界面：简约美观的登录界面，用户需输入账号密码到已设有的输入框，待平台系统核对后用户方可进入主页。如果用户忘记了密码，此界面备有忘记密码的功能协助用户找回密码。如果用户未曾注册过本平台便可点击此界面的注册按钮进图对应的界面进行注册。如图 6.1。

注册页面：此页面提供未曾注册本平台的用户注册用户账号。用户需要输入完整信息以便完成注册。此外，用户可以选着是普通用户还是卖家用户。因为卖家的注册与普通用户类似，所以本文便设买家的注册功能在与此处。如图 6.2。



图 6.1 登录界面

图 6.2 注册界面

忘记密码页面：用户可通过手机号或邮箱验证找回密码或修改密码。如图 6.3。一键登录页面：考虑现在大多数的移动软件都配备一键登录的按钮让用户能快速通过登录，因此本软件平台也设有通过输入手机号直接登录绑定的账号。如图 6.4。

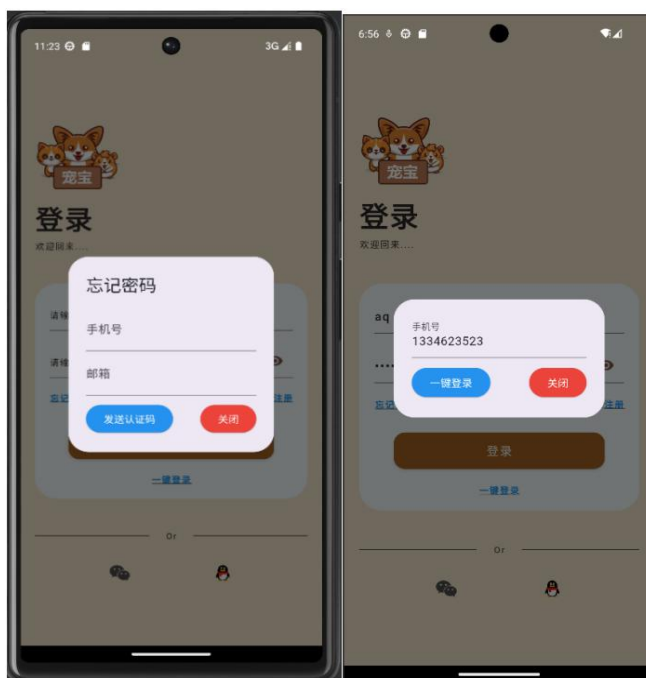


图 6.3 忘记密码界面

图 6.4 一键登录界面

主页页面：如图所示用户可以在主页里选择试养、寄宿、美容、医院、商城等服务功能，进入对应界面。页面底部配有底部栏功能让用户点击进入发现、消息、购物车、我的页面。而页面中部提供了帖子专区，分别有关注和推荐的贴纸提供给用户选择查看。用户可以点击感兴趣的帖子进入对应帖子的详细页面。如图 6.5。关注推荐页面：用户点击帖子就能进入到对应帖子的详情页面，如图所示。如图 6.6。在本界面底部设有上一章节表 5.2 所提到的底部导航 `BottomNavigationView()`。`BottomNavigationView()`使本文自定义编写的一个外部类，其中包含了首页、发现、消息、购物车、我的等按钮。



图 6.5 主页界面



图 6.6 关注推荐帖子界面

发现页面：用户点击了主页页面的底部功能栏里的发现按钮就会进入如图所示的页面，此页面提供用户观看短视频，用户可以点击关注、推荐、本地进入对应的视频频道，本文也有借鉴著名的视频软件平台抖音的使用方式。用户可以滑动来实现跳转到关注、推荐和本地的页面。此外，用户可以喜欢的短视频进行点赞、收藏、分享。如图 6.7。

消息页面：此页面提供用户与用户或者卖家之间的交流以及接受系统或者商家的消息。如图 6.8。本软件平台也对用户发送或接受的消息进行了处理和分类以方便用户和避免用户因过多的消息而忽略了一些重要的消息。

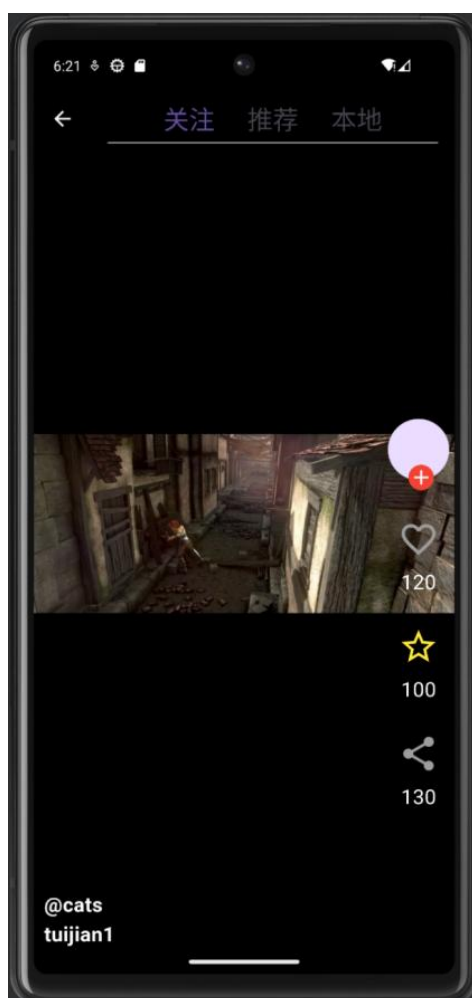


图 6.7 发现界面



图 6.8 消息界面

购物车页面：此页面显示用户先前加入购物车的物品。此页会显示用户所加入的商品的数量、价钱、商品名和其店家名。当用户决定要购买商品或者服务后可以点击选择在该商品和服务的圆框按钮。接着点击结算按钮就会跳到购物车结算并为用户结算所需支付的费用。在此界面的右上方有个管理按钮是提供给用户管理购物车的按钮。用户可以对购物车里的商品进行批量的处理如删除、选定等。如图 6.9。支付成功页面：此页面是当用户支付完商品或者服务后并且支付成功后的页面如图 6.10。



图 6.9 购物车界面



图 6.10 支付成功界面

我的页面：此页面是提供用户一些管理个人账户的页面。首先，用户可以看到用户的信息，同行里也设有 AI 客服、设置页面按钮提供用户点击。下两行设有各类按钮提供用户以查看自己收藏，关注店铺、足迹、零钱，还可以查看用户的订单状态。如图 6.11。客服页面：此页面是提供用户质询一些常见的问题。首先，用户会进入到设有到常见的问题的页面，通过点击对应的问题，客服会自动回复，如此能够更快地帮助用户解惑，如果用户的困惑不存在常见问题列表里用户可以点击在线客服，提出问题，人工客服会回答用户的问题。如图 6.12。



图 6.11 我的界面

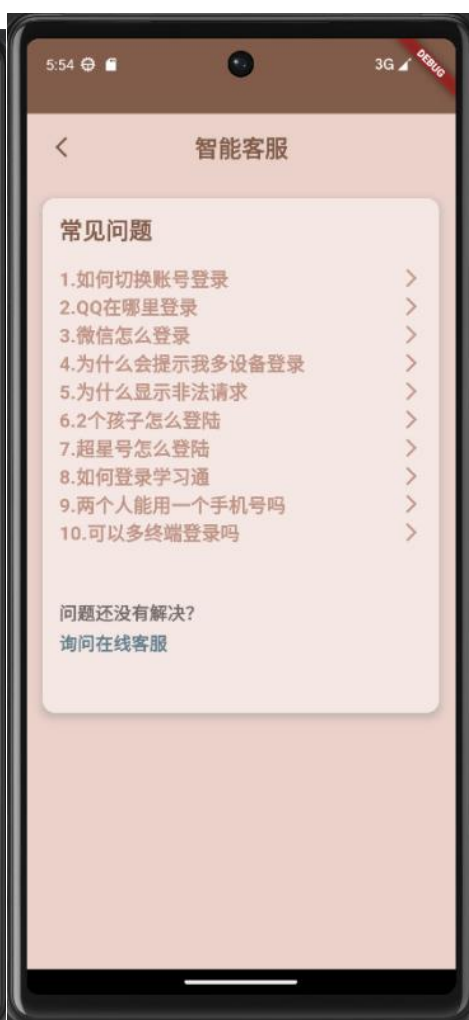


图 6.12 客服常见问题界面

设置界面：此页面是提供用户进行账号管理、切换地址、通知、通用、帮助、邀请好友和退出登录功能。如图 6.13。



图 6.13 设置界面

收藏页面：用户进入收藏页面可以查看自己收藏的宝贝，店铺，图文，视频。如图 6.14、图 6.15。



图 6.14 收藏-宝贝界面 图 6.15 收藏- 店铺界面

关注店铺页面：进入到关注店铺用户可以查看自己关注的店铺，常逛的店铺还有买过商品的店铺。如图 6.16、图 6.17。



图 6.16 关注店铺-关注界面 图 6.17 关注店铺-常逛界面

足迹页面：进入足迹页面，用户便可以查看自己浏览过的商品，店铺，图文，视频。方便用户查看自己浏览记录。如图 6.18、图 6.19。



图 6.18 足迹-宝贝界面

图 6.19 足迹-店铺界面

我的零钱页面：此页面是平台假设的一个电子钱包，用户可以充钱放入钱包，还可以查看自己用户账单。此页面本文是借鉴淘宝的零钱界面。如图 6.20。市面上许多软件平台都有了电商软件的功能，因此本软件平台也设置了许多电商功能。

账单页面：进入账单，用户可以查看自己的收付款记录，包括时间，商家等。如图 6.21。



图 6.20 我的零钱界面



图 6.21 账单界面

订单状态页面：用户进入订单状态页面可以查看商品处于的状态如待付款，待收货，待发货，退款，待评价。这也是电商平台必有的功能以及界面之一如图 6.22、图 6.24。这可以使得用户方便于追踪商品的状态。此界面滑动到底部时，平台也会推荐一些热门商品服务等给用户选择。如果用户对商品和服务不满意也能点击退款/售后按钮进行退款。而右上角的筛选则是方便用户对商品进行批量管理等操作。



图 6.22 全部订单界面



图 6.23 待付款界面

试养页面：此页面是本软件平台的最重要和亮点之一。此页面本软件平台会推荐显示一些宠物试养店的信息。试养首页分类展示了各种宠物的情况和视频。用户选择了心仪宠物，可以查看该宠物更为详细的信息。用户决定试养该宠物后点击底部的开始试养就能填写预约信息和时间便可试养该宠物。如图 6.25、图 6.26。



图 6.25 试养主页



图 6.26 试养宠物详情界面

寄宿页面：寄宿首页会展现本软件平台推荐的寄宿店家，用户可以在首页选择优惠商品或进入寄宿店详情界面，进入寄宿店后可购买该家店的商品，或预约寄养服务，同时还能查看店铺的环境和评价。如图 6.27。



图 6.27 寄宿主页界面

美容页面：美容界面首页会显示一些本软件平台推荐的美容店家。用户可以在此页选择宠物的美容护理服务或进入美容店详情界面，进入美容店后可购买该家店的商品，或预约服务，同时还能查看店铺的环境和评价。此外本人就普通还提供了搜索栏给用户进行搜索想要的商品和服务。界面中间也设有许多最近热门优惠的商品或服务如图 6.28，图 6.29。



图 6.28 美容主页界面



图 6.29 美容店家界面

医院页面：宠物医院界面首页会显示一些本软件平台推荐的宠物医院。用户可以在此页选择购买药品或进入医院详情界面，进入医院后可购买该家医院的药品，或预约服务，同时还能查看医院的环境和医师以及评价。如图 6.30、图 6.31。



图 6.30 医院主页界面



图 6.31 医院详情界面

商城页面：此页面用户可选择需要购买的商品，点击后进入商品详情界面查看商品详情并加入购物车，也可以搜索商品。如图 6.32、图 6.33。



图 6.32 商城 1 界面



图 6.33 商品详情界面

训练页面：此页面提供一系列教学视频给用户观看学习，用户可以从中增加自己的实验经验。右侧一列则是一系列课程分类提供给用户进行选择查看想要学习的视频。而在此页面还提供了用户选择根据种类、视频或者技能和时长分类。如图 6.34。



图 6.34 训练界面

宠物日志页面：用户可以记录宠物每天的生活日常、宠物复诊时间、宠物的成长等。用户可以设置很多日志来提醒所饲养的宠物需要健康检查了、体检了、身体状态、心情等状态。设置好后，平台也会通过手机通知信息提醒用户。如图 6.35、图 6.46。



图 6.35 宠物日志界面



图 6.36 宠物日志添加日志界面

我的宠物页面：用户可以分享自己宠物的信息。此外用户可以记录宠物的成长记录。如图 6.37。

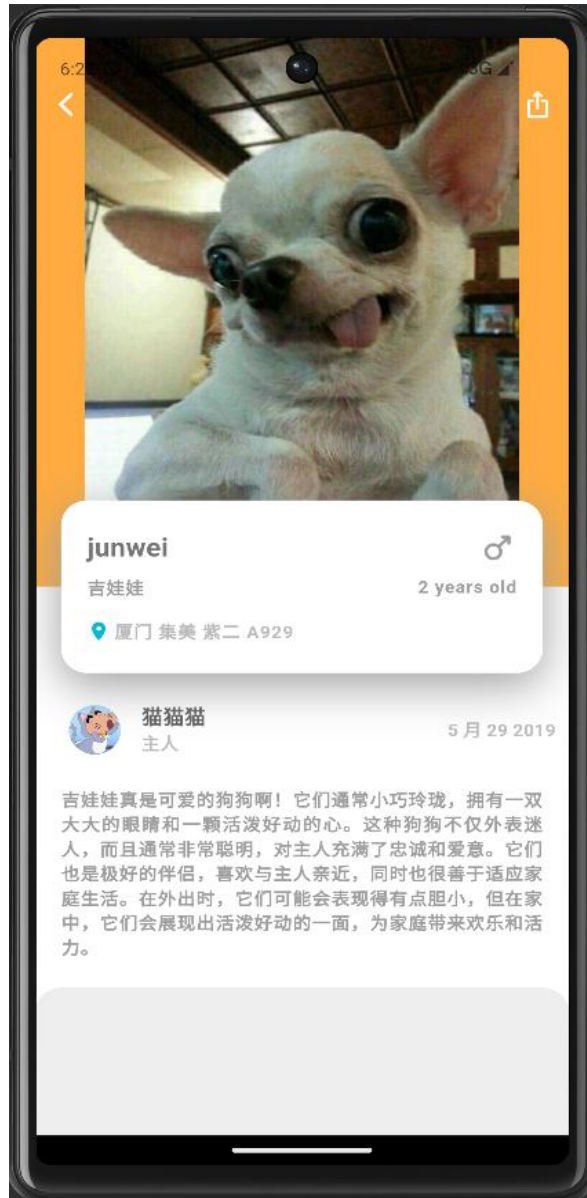


图 6.37 我的宠物界面

6.3 平台安装部署及步骤

从经济，实用，方便性等因素，本文采用极为的安装方法将《宠宝》安装至安卓手机上，就是直接导出项目成 apk 在接着导入到安卓手机里。首先，在 Android studio 里的控制台里输入 Flutter build apk。之后，AndroidStudio 就会生成一个 apk 和其所在的路径。接着，根据此路径就能提取已打包好的 apk，再把它输入进需安装的安卓手机里即可。图 6.38 显示本台成功按照生成路径找到了所导出的 apk 并成功地导入到安卓移动端上。

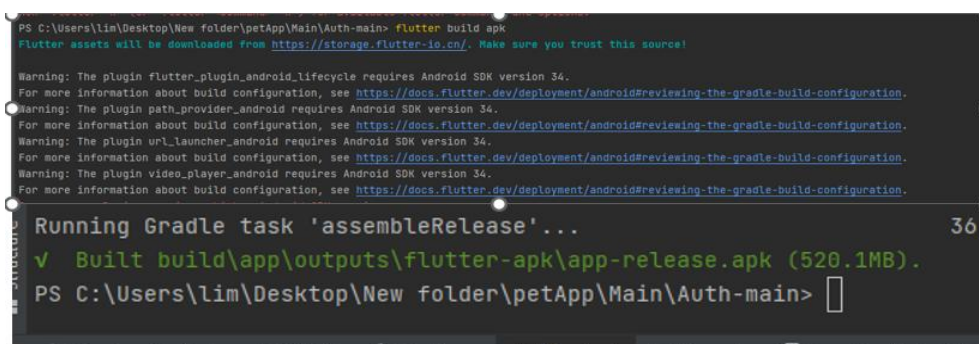


图 6.38 导出 apk 安装包

以下图片即图 6.39、图 6.40、图 6.41 是移动端安装本平台软件的步骤。在图 6.42 则可以看出，本平台系统运行在 Android 手机里和在 Android Studio 虚拟机里的结果完全一致。

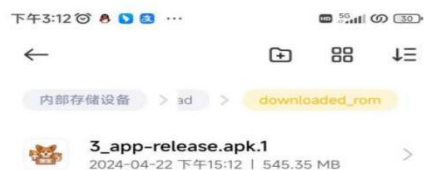


图 6.39 apk 成功导入手机



图 6.40 安装 apk

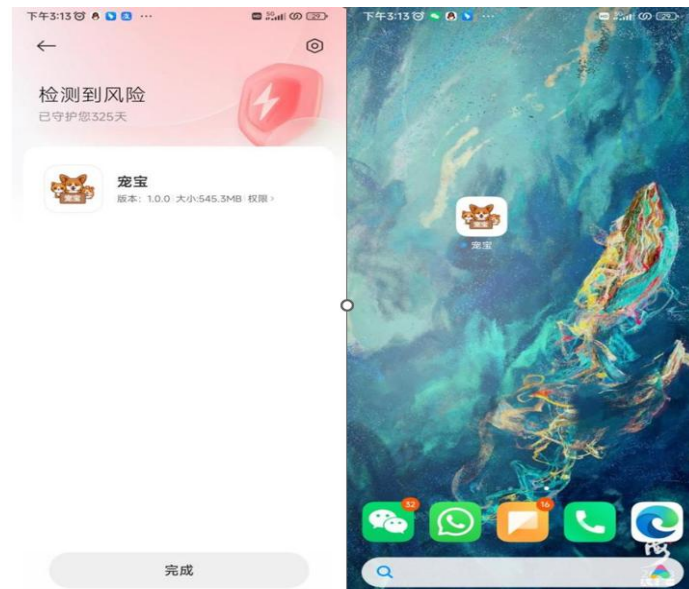


图 6.41 成功安装

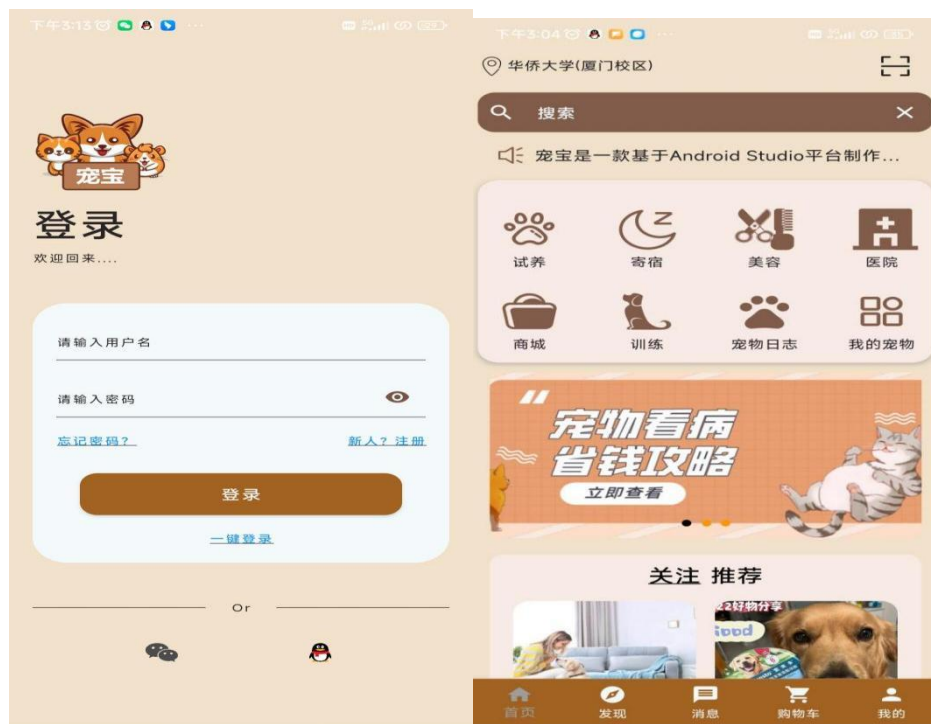


图 6.42 手机运行 apk

6.4 本章小结

本章节展示了本平台的进行了黑盒测试和功能测试，并介绍了本平台在虚拟机上运行的效果图。接着，本章将平台导出成 **apk** 手机安装包并成功安装在安装易懂端上。

第七章 总结和展望

本文旨在针对日益严重的弃养现象提出解决或减缓措施，并探讨通过宠物软件平台来应对该问题。本文设计了宠物试养软件平台《宠宝》，引入了“试养”概念。即在人们饲养宠物前，他们可以通过一段时间的试养来确认自己是否已做好充分的准备和条件，从而大大减少因一时兴起而弃养宠物的现象。

接下来，本文介绍了如何实现该平软件平台的过程。本文是使用 Flutter UI 框架+Dart 语言+Spring Boot+Mysql 等技术完整地设计实现该软件平台。Flutter UI 框架是近年来刚上市便非常热门的移动端设计开发平台。其设有的热重载功能深得开发人员得喜爱，因其具备能在不重启程序的情况下实时更新显示代码和结果，这大大缩减了开发人员的开发时间以及成本。

此外，《宠宝》拥有非常丰富各式各样的功能提供给用户使用，其中最核心的功能是试养。其余的有宠物寄宿，宠物医院，宠物美容等功能大大减轻了用户即饲主在饲养宠物过程中的负担。用户可以使用一款手机应用轻松完成多项任务，如购买宠物用品、预约宠物医院、美容服务、宠物寄宿等，同时还能与其他用户交友聊天，观看有关宠物饲养的短视频等。

由于时间和技术存在限制，本软件平台《宠宝》还存在一些不足之处。由于本软件平台并没有大面积推广使用，导致了测试用例不足，因而本软件平台可能存在一些性能隐患，像是数据是否实时更新，正确性以及安全。因此透过接下来的工作去不断完善及维护，来提高本软件平台的质量。

参考文献

- [1] 刘冰洁. 智能宠物寄宿系统研究与设计[J]. 信息记录材料, 2021, 22(01): 199-201. DOI: 10.16009/j.cnki.cn13-1295/tq.2021.01.137.
- [2] 颜惠. 基于 Web 的宠物店信息管理系统设计[J]. 软件, 2023, 44(02): 147-149.
- [3] 朱建波, 刘海涛, 李雷川, 等. 跨平台框架 Flutter 在新闻 APP 中的应用研究[J]. 中国报业, 2022(09): 40-41. DOI: 10.13854/j.cnki.cni.2022.09.050.
- [4] 文淑华, 黄明源, 刘雪帆等. 基于 Flutter 和 SpringBoot 的多平台会议室管理系统[J]. 实验科学与技术, 2021, 19(01): 155-160.
- [5] 罗光武, 陈典灿, 吴荷, 等. 应用 Springboot+Vue 框架的时间管理软件的设计与实现[J]. 工业控制计算机, 2024, 37(04): 64-66.
- [6] 宋刘杰. 宠物犬的饲养管理及疫病防治[J]. 河南农业, 2023(28): 57. DOI: 10.15904/j.cnki.hnny.2023.28.024.
- [7] 王英明, 许青, 郭艳梅. 基于 Flutter 的智慧工地 APP 的设计与实现[J]. 通化师范学院学报, 2022, 43(04): 117-121. DOI: 10.13877/j.cnki.cn22-1284.2022.04.018.
- [8] 舒红梅, 符清芳, 张艳等. Flutter 框架在前端开发中的应用探析[J]. 软件, 2022, 43(03): 55-58.
- [9] 庄帅. 宠物管理系统的设计及实现[J]. 福建电脑, 2022, 38(05): 68-72. DOI: 10.16707/j.cnki.fjpc.2022.05.017.
- [10] 张静, 段修军, 张祥光. 城市宠物犬行为训练[C]//中国畜牧兽医学会犬业分会, 公安部南昌警犬基地. 第 18 次全国犬业科技学术研讨会论文集.[出版者不详], 2018: 4
- [11] 胡瑚. 基于 MySQL 的科研信息管理系统数据库设计[J]. 信息与电脑(理论版), 2024, 36(04): 169-171.
- [12] 李志伟. 基于 Android Studio 开发平台的简易阅读器设计与实现[J]. 电脑编程技巧与维护, 2023(09): 47-49. DOI: 10.16184/j.cnki.comprg.2023.09.049.
- [13] T.Bailey, Alessandro Biessek, Trevor Wills. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart[M]. Packt Publishing, 2021. [2023-05-21]. <http://ieeexplore.ieee.org/document/10162491>.
- [14] S.Alessandria, Brian Kayfitz. Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart[M]. Packt Publishing, 2021. [2023-05-21]. <http://ieeexplore.ieee.org/document/10163215>.
- [15] R.Nagy. Simplifying Application Development with Kotlin Multiplatform Mobile: Write robust native applications for iOS and Android efficiently[M]. Packt Publishing, 2022. [2023-05-21]. <http://ieeexplore.ieee.org/document/10162643>.

- [16] K. Kishore, S. Khare, V. Uniyal, S. Verma. Performance and stability Comparison of React and Flutter: Cross-platform Application Development[C]//2022 International Conference on Cyber Resilience (ICCR). 2022: 1-4. [2023-05-21].
- [17] R. Mamoun, M. Nasor, S. H. Abulikailik. Design and Development of Mobile Healthcare Application Prototype Using Flutter[C]//2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). 2021: 1-6. [2023-05-21]. DOI: 10.1109/ICCCEEE49695.2021.9429595.

致谢

时光如白驹过隙，转瞬即逝，已经到了本科生即将毕业之际，在华侨大学的四年本科生涯中，我度过了非常快乐且充实的时光，收获颇丰。尽管这段时光中被突如其来的疫情打乱了节奏，让我无法和大家共同度过其中的两年，但我依然感激这段经历所带来的成长和体验。我也从中感受到了老师的关心，同学的热情帮助，让我体验到了满满的人情味，我非常感激和激动即使我只能在远程线上与大家交流。所幸的是，在最后的两年里，我能返校与大家一起共度，以及参加我的毕业论文。在此在向所有关心和帮助我的老师们、同学们、学长们和朋友们表示感谢。

首先，我要向我的导师萧宗志副教授表示由衷地致谢，感谢他从头到尾都给予非常多的耐心老协助我撰写我的毕业论文包括了选题、软件开发过程等等方面的帮助。他给予了我许多宝贵的思路和建议。此外在我返校后，萧老师常常带我做项目，参观企业等，帮我增长了不少的见识，也是我人生学习的黄金时期。特此感谢我的导师，萧老师，正是因为他，我才能顺利完成我的论文。

其次，我要感谢华侨大学的计算机科学与技术学院为我提供了一个良好的学习环境，设备齐全的实验室，使得我能好好专心地完成我的毕业论文及项目。

此外，我要感谢我的室友罗文明，我的朋友郭光顺、洪宥维，在这期间互相监督我和互相鼓励，并给我很大的鼓励和帮助。

最后，我要感谢我的家人，他们对于我学业上的支持和关心，让我能能继续努力，不辜负他们。