# 余旺

# 7.1

## 项目进展情况

熟悉了最基础的安卓开发使用，完成了最基础的登录注册界面，对前端页面控件有了一定的了解。

## 工作问题及重点关注

页面跳转：需要掌握活动（Activity）之间的跳转和数据传递。

页面布局：需要了解和掌握布局方式，如线性布局（LinearLayout）、相对布局（RelativeLayout）和约束布局（ConstraintLayout）。

美观设计：为了设计出好看的前端界面，需要进一步学习和运用Material Design规范，掌握更多UI组件的使用。

## 工作分析与收获

基础前端控件：通过对安卓前端控件的学习，了解了TextView、EditText、Button等基础控件的使用。

登录注册界面：完成了登录注册界面的基本功能，实现了用户输入信息的验证和提交。

开发技巧：积累了一些安卓开发的技巧，如如何调试、如何使用安卓布局编辑器等。

## 下一步计划

深入学习页面跳转和布局：通过更多的实例和项目练习，熟悉Activity之间的交互和不同布局的使用。

美化前端界面：学习和应用Material Design规范，利用颜色、阴影、动画等元素提升界面的美观度。

优化用户体验：在已有的登录注册界面基础上，增加用户体验优化，如输入框的提示、错误信息的展示等。

# 7.2

## 项目进展情况

登录注册功能进行完善，完成了页面的跳转，界面进行了优化，同时对网络通讯辅助组员共同开始，对flask进行学习

## 工作问题及重点关注

网络通讯：需要深入学习和掌握如何在Android中进行网络请求和数据传输，这一块需要重点关注。

如何设计优美的界面仍待学习

## 工作分析与收获

基础前端控件：通过对安卓前端控件的学习，了解了TextView、EditText、Button等基础控件的使用。

登录注册界面：完成了登录注册界面的基本功能，实现了用户输入信息的验证和提交。

开发技巧：积累了一些安卓开发的技巧，如如何调试、如何使用安卓布局编辑器等。同7.1进一步进步了。

网络通讯：开始研究网络通讯的相关知识，辅助组员共同学习Flask，为后端服务的开发打下基础。

# 7.3

## 项目进展情况

界面的初始设计已经完成，供小组成员进行实验测试，同时辅助完成网络通信连接

## 工作问题及重点关注

组件：对于一些组件有了更加深入的了解，在7.3晚上花费了大量时间卡在了布局上，最终通过权重设置的方法成功解决

网络通信：基础的登录注册通信传输的完成让我们能够继续开展后期的定位功能等

## 工作分析与收获

深入学习页面跳转和布局：通过更多的实例和项目练习，熟悉Activity之间的交互和不同布局的使用，特别是解决按照权重布局的问题。

美化前端界面：学习Material Design规范，利用颜色、阴影、动画等元素提升界面的美观度，这一块还在学习，

优化用户体验：在已有的登录注册界面基础上，增加用户体验优化，如输入框的提示、错误信息的展示等。

网络通讯的实现：学习如何在Android中进行网络请求（使用Retrofit或OkHttp），并与Flask后端进行数据交互，完成基本的登录注册网络通讯功能

# 7.4

## 项目进展情况

主界面完成，但在答辩后决定优化界面，最终废弃掉了该版本，往更加美观的版本出发，开始尝试拖拽控件进行操作而非直接以代码形式完成前端的设计

## 工作问题及重点关注

界面设计工具：开始使用拖拽控件进行操作，尝试通过UI设计工具而不是直接以代码形式完成前端设计，需要适应新的工作方式并解决可能遇到的问题。

## 工作分析与收获

适应新工具：通过拖拽控件进行操作，提升了设计界面的效率，同时也需要克服新的学习曲线。

开发技巧：积累了一些安卓开发的技巧，如如何调试、如何使用安卓布局编辑器等。

# 7.5

## 项目进展情况

在开发过程中，我主要学习和掌握了RecyclerView和RatingBar等布局的使用。通过大量的实践操作，我成功地将这些新学到的控件应用到了项目中，并且实现了一个功能完整的模板。这一过程中，我不仅了解了这些控件的基础用法，还熟悉了它们的高级特性和自定义方法。

## 工作问题及重点关注

在应用RecyclerView和RatingBar的过程中，遇到了一些问题和挑战。首先是RecyclerView的布局管理和适配器的使用。由于数据源的动态变化，如何高效地更新和管理RecyclerView成为一个关键问题。其次是RatingBar的样式和交互设计，需要确保用户体验的流畅性和视觉效果的一致性。为了解决这些问题，我参考了大量的文档和示例代码，并进行了多次调试和优化。

重点关注的方面包括：

RecyclerView的性能优化，特别是在大量数据加载时的流畅度。

RatingBar的自定义样式，确保其在不同分辨率和设备上的一致表现。

用户交互的细节处理，包括点击事件的响应和动画效果的实现。

## 工作分析与收获

通过本周的学习和实践，我对RecyclerView和RatingBar有了更加深入的理解。RecyclerView作为一个强大的列表控件，其灵活性和可扩展性让我能够实现复杂的列表布局和交互效果。而RatingBar作为评分控件，其简单易用的特性使得用户可以直观地进行评分操作。在工作中，我收获了以下几点：

技术提升：掌握了RecyclerView和RatingBar的使用技巧，并能够根据项目需求进行自定义。

问题解决能力：通过解决实际开发中遇到的问题，提高了我的调试和问题排查能力。

项目经验积累：将所学知识应用到实际项目中，完成了一个功能完整的模板，为后续的开发打下了坚实的基础。

# 7.6

## 项目进展情况

· 界面整体开发完成：已完成两个主界面的开发，并优化了界面设计。

· 底部导航功能完成：成功实现底部导航，能够在不同的主界面之间进行切换。

· 用户交互体验优化：改进了用户界面，提升了用户体验

## 工作问题及重点关注

· 界面布局问题：在开发过程中，主要挑战是在保证界面美观和功能性的前提下进行布局优化。

· 页面跳转与状态管理：确保在不同页面之间切换时，状态能够正确保留和恢复。

· 性能优化：需要关注应用在不同设备上的性能，确保流畅运行，这一方面在后期功能合并等任务出现的时候需要去解决和研究

## 工作分析与收获

· 技术提升：

深入理解了Android开发中的界面布局、页面跳转和状态管理。

熟悉了使用Fragment实现底部导航的技术。

· 问题解决能力：

面对界面布局的挑战，通过不断尝试和调整，最终找到了合适的解决方案。

在页面跳转和状态管理上，学会了如何在不同Fragment之间传递数据和保持状态。

· 团队合作与项目管理：

在与团队成员的合作中，学会了如何有效地沟通和协作，提高了项目的开发效率。

通过项目管理工具，合理安排任务，确保项目按时完成。

· 用户体验设计：

在界面设计中，更加注重用户体验，学会了如何通过细节优化提升用户满意度。

# 7.7

## 项目进展情况

成功利用 RecyclerView 模仿美团、饿了么等设计出界面，界面效果美观简洁。用户可以在主界面上查看不同类别的食品，点击类别时会更新显示对应的食品列表。

通过结合静态和动态 RecyclerView，实现了界面内容的动态更新，提升了用户体验。

## 工作问题及重点关注

RecyclerView 性能优化：在大量数据加载时，可能会出现卡顿现象，需要进一步优化 RecyclerView 的性能。如何通过使用 ViewHolder 模式、RecyclerView 缓存、避免过多的视图嵌套等方式提高性能。

数据加载与更新：确保数据加载和更新的效率，特别是在用户频繁切换类别时，数据的快速加载和展示。

界面交互优化：确保用户在操作时界面响应及时，提供平滑的动画效果和反馈。

错误处理与稳定性：确保应用在各种操作和异常情况下的稳定性，处理可能的崩溃和错误。

## 工作分析与收获

· 熟悉 RecyclerView：通过本次开发，进一步熟悉了 RecyclerView 的使用和自定义 Adapter 的开发，对 RecyclerView 的性能优化有了更深的理解。

· 提升 UI 设计能力：通过模仿美团、饿了么等应用的设计，提升了界面设计和用户体验方面的能力，学会了如何通过简洁美观的设计吸引用户。

· 增强问题解决能力：在开发过程中遇到了一些技术问题，通过查阅文档、在线搜索和同事讨论，成功解决了问题，提升了问题解决的能力。

# 7.8

## 项目进展情况

解决了动态Model回收展示的相关Bug：

成功解决了Recyclerview中动态Model的回收和展示问题，使图片能够成功显示。

主页面和导航页面的设计与美化：

设计并美化了主页面和导航页面的整体框架。

加入了GIF图、FloatingActionsMenu、底部导航栏等组件，增强了界面的互动性和美观度。

界面跳转和接口设计：

为界面跳转和接口设计预留了空间，并进行了部分填补。

## 工作问题及重点关注

确保动态Model在不同设备上的一致性展示。

继续优化界面美观度和用户体验。

完善界面跳转和接口设计，确保流畅的用户操作体验

## 工作分析与收获

通过解决Recyclerview的动态Model回收问题，深入了解了Android中Recyclerview的工作原理。

在设计和美化界面的过程中，学会了如何有效地使用各种UI组件，使应用更加吸引用户。

对于项目中出现的Bug，学会了如何进行调试和解决，提高了问题解决能力。

# 7.9

## 项目进展情况

· 商品评价界面的开发：

完成了商品评价界面的初步设计和实现。

界面功能包括展示评价、提交评价、评分等。、

· 优化界面多组件卡顿的问题：

通过代码优化和性能调优，显著减少了界面卡顿现象。

· 加入动画效果：

在界面切换、按钮点击等操作中加入了动画效果，提升用户体验。

· 协助开发服务器的数据传输：

参与设计和实现了服务器与客户端的数据传输接口。

· 相关文档绘图，对于队员代码进行合并，代码测试

## 工作问题及重点关注

· 界面优化仍有待提高：

尽管界面卡顿问题有所改善，但在复杂界面和大量数据展示时仍有优化空间。

· 动画效果的流畅性：

部分动画效果在低性能设备上表现不佳，需要进一步优化

。

· 数据传输的稳定性：

在高并发环境下，数据传输的稳定性和可靠性仍需重点关注和测试。

· 用户体验：

需要持续收集用户反馈，并根据反馈不断优化界面和功能。

## 工作分析与收获

· 技术提升：

通过解决界面卡顿问题，掌握了更多的性能优化技巧和方法。

学习并使用了动画框架，提升了界面的交互性和视觉效果。

参与服务器数据传输的开发，增强了对客户端与服务器交互的理解。

· 团队协作：

在与团队成员的协作过程中，提升了沟通和协作能力。

学会了如何有效地分配任务和资源，保证项目的顺利进行。

· 问题解决：

在解决实际问题的过程中，积累了丰富的经验，提升了问题分析和解决能力。

通过持续的测试和优化，提高了项目的质量和稳定性。

· 用户体验：

重视用户反馈，通过不断优化和改进，提升了用户满意度。

通过对用户行为的分析，找到了界面和功能的改进方向。

# 7.10

## 项目进展情况

· 修改整体界面：对项目的整体界面进行了美化和优化，包括调整界面布局和比例，提升用户体验。

· 登录注册功能：在登录注册界面中加入了“记住密码”功能，方便用户下次登录。

· 代码合并：成功合并了团队成员的代码，确保了各模块的集成和兼容。

· 登录登出服务器功能：协助团队完成了与服务器的登录登出功能，实现了用户信息的存储与管理。

· 数据库的设计与完善

## 工作问题及重点关注

· 界面适配：在不同屏幕尺寸上的界面适配存在一定问题，需进一步优化。

· 记住密码功能：需要确保记住密码功能的安全性，防止用户信息泄露。

· 代码合并冲突：在合并代码过程中出现了一些冲突，需关注代码管理和团队协作问题。

· 服务器通信：需要重点测试服务器通信的稳定性和可靠性，确保登录登出功能的顺畅。

## 工作分析与收获

· 界面优化：通过对界面的优化和调整，提升了用户体验，增强了项目的美观度和实用性。

· 功能实现：加入记住密码功能，提高了用户的便捷性，同时也积累了在用户信息管理方面的经验。

· 团队协作：通过代码合并和功能协作，提升了团队的协作能力，增强了项目的整体进度和质量。

· 服务器功能：在实现登录登出服务器功能的过程中，深入理解了客户端与服务器之间的通信原理，提升了技术水平。

# 7.11

## 项目进展情况

**1. 界面风格统一**

色彩与品牌一致性：首先确定与品牌调性相符的主色调和辅助色，确保整个应用或网站在视觉上保持一致性。同时，使用清晰的色彩对比来提高信息的可读性。

字体与排版规范：选择易于阅读的字体，并设定统一的字号、行间距和段落间距，确保在不同设备上都能保持良好的阅读体验。

图标与图片标准化：设计或选用高质量的图标和图片，确保它们在不同分辨率下都能清晰显示，同时与整体风格相协调。

**2. 评论与商家界面整合**

信息架构优化：重新设计信息架构，将评论与商家信息以更直观、紧密的方式整合在一起。例如，在商家详情页中嵌入滚动评论区，用户无需跳转即可查看和发表评论。

交互流畅性：设计流畅的交互逻辑，使用户能够轻松地在商家信息与评论之间切换。提供评论筛选、排序等功能，以及快速回复评论的按钮，提升用户体验。

**3. 界面布局调整与自适应设计**

网格系统应用：运用网格系统规划界面布局，使元素排列有序、空间分配合理。通过百分比或视口单位定义元素宽度和间距，实现布局的自适应调整。

测试与调整：在多种设备和分辨率上进行测试，包括不同品牌和型号的智能手机，确保界面在不同设备上都能良好工作。根据测试结果进行必要的调整和优化。

**4. 登录注册流程调整**

简化流程：减少不必要的步骤和字段，只收集关键信息。提供快速登录选项，如社交账号登录、手机号验证码登录等，提升用户注册和登录的便捷性。

增强安全性：加强密码强度验证、短信验证码验证等安全措施，确保用户信息安全。同时，提供密码找回和账号保护等功能，增强用户信任感。

个性化体验：在登录或注册成功后，根据用户信息提供个性化的推荐内容或优惠信息，提升用户粘性和满意度。

## 工作问题及重点关注

**技术实现难度**：界面优化和整合可能涉及复杂的前端技术实现，需要关注技术团队的能力和经验是否足够应对。

**用户体验反馈：**优化后的界面是否真正提升了用户体验？需要密切关注用户反馈，及时调整和优化。

**项目时间管理**：界面优化和整合可能需要较长的时间来完成，需要合理规划项目时间，确保按时交付。

**资源分配**：在资源有限的情况下，如何合理分配资源以支持界面优化和整合工作？需要关注资源的使用效率和优先级设置。

## 工作分析与收获

***技术能力提升***：通过界面优化和整合工作，团队成员在前端技术、用户体验设计等方面的能力得到了提升。

**用户体验优化**：优化后的界面更加直观、易用，提升了用户的满意度和忠诚度。同时，通过用户反馈的收集和分析，团队对产品有了更深入的了解和认识。

**项目经验积累**：本次界面优化和整合工作为团队积累了宝贵的项目经验，为未来的产品开发和优化提供了有力的支持

# 7.12

## 项目进展情况

**评价功能完善：**用户评价系统已经全面优化，不仅支持文字评价，还增加了图片、丰富了用户反馈的维度。同时，评价系统也加强了内容审核机制，确保评价的真实性和合法性。

**登录注册流程优化**：登录注册界面和功能得到了进一步完善，特别是指纹识别功能的加入，大大提升了用户登录的便捷性和安全性。此外，记住密码功能也得到了优化，确保用户在下次访问时能够快速登录，无需重复输入账号密码。

**验证是否登录的跳过界面完善**：针对已登录用户，系统设计了智能的跳过登录验证界面功能，根据用户的登录状态自动调整页面显示，减少了用户的操作步骤，提升了用户体验。

**聊天功能：**用户的聊天界面进行设计，和实现，但遇到了界面设计困难等问题

## 工作问题及重点关注

**用户隐私保护：**随着指纹识别等生物识别技术的引入，用户隐私保护成为了一个重要问题。我们需要确保所有用户数据的安全存储和传输，防止信息泄露。

**兼容性测试**：不同品牌和型号的手机在硬件和软件方面存在差异，可能导致部分功能在某些设备上无法正常运行。因此，我们需要进行全面的兼容性测试，确保项目在所有目标设备上都能良好工作。

**用户反馈处理**：随着项目的推进，用户反馈将逐渐增多。我们需要建立有效的用户反馈机制，及时收集和处理用户反馈，确保项目能够持续优化和改进

## 工作分析与收获

技术实力提升：在项目实施过程中，团队成员通过不断学习和实践，技术实力得到了显著提升。特别是在前端交互、后端逻辑处理以及安全性保障等方面积累了宝贵的经验。

用户导向思维：本次项目让我们更加深刻地认识到用户导向的重要性。只有真正关注用户需求、提升用户体验，才能赢得用户的信任和支持。因此，在未来的工作中，我们将继续秉承用户导向的思维模式，不断优化产品和服务

项目管理经验积累：通过本次项目的实施和管理，我们积累了宝贵的项目管理经验。包括如何制定项目计划、如何分配资源、如何监控进度以及如何处理风险等。这些经验将为我们未来的项目管理工作提供有力的支持。

# 7.13

## 项目进展情况

在最近的项目开发周期中，我们进行了多项功能尝试与调整。首先，我尝试引入了聊天功能，旨在为用户提供更加丰富的交互体验。然而，在经过深入的评估和实际测试后，我发现聊天功能的实现需要投入大量的资源，并且在当前项目的定位和市场需求下，其优先级相对较低。因此，出于实际考虑，我们决定放弃该界面的设计和聊天功能的实现。

随后，我将注意力转向了语音识别和文字转语音这两项技术。经过团队的共同努力，我们成功地将这两项功能集成到项目中，并在虚拟机环境中进行了初步测试，结果显示功能运行稳定且符合预期。然而，在将应用部署到安卓设备进行进一步测试时，我们遇到了问题：语音识别和文字转语音功能在安卓设备上无法正常使用。这一发现让我们意识到，虽然功能在开发环境中表现良好，但在实际部署时仍需考虑不同平台的兼容性和性能差异。

## 工作问题及重点关注

当前我们面临的主要问题是语音识别和文字转语音功能在安卓设备上的兼容性问题。这要求我们重点关注以下几个方面：

**平台兼容性**：深入分析安卓系统对于语音识别和文字转语音技术的支持情况，找出导致功能无法使用的具体原因。

**性能优化**：针对安卓设备的特性，对语音识别和文字转语音功能进行性能优化，确保在安卓平台上能够流畅运行。

**替代方案评估：**如果当前使用的技术栈在安卓平台上难以实现预期效果，我们需要评估并考虑采用其他替代方案。

## 工作分析与收获

**技术选型的重要性：**在项目初期，对技术选型的慎重考虑至关重要。我们需要充分考虑技术的可行性、兼容性以及未来扩展的可能性，避免在项目后期出现无法克服的技术难题。

**灵活调整项目计划：**面对项目中的不确定性和挑战，我们需要保持灵活性，根据项目实际情况及时调整项目计划。这有助于我们更好地应对变化，确保项目的顺利进行。

**深入学习与技能提升**：此次项目让我们深刻认识到技术更新的迅速和复杂性。为了保持竞争力，我们需要不断学习新技术、新工具，提升自己的专业技能和综合素质。

# 7.14

## 项目进展情况

**打电话功能开发：**

成功完成了打电话功能的开发工作。该功能允许用户在应用内直接发起电话呼叫，无需跳转到系统拨号界面，提升了用户体验。。

**项目测试与Debug：**

随后，我们对整个项目进行了全面的测试和Debug工作。测试覆盖了所有已开发的功能模块，包括新开发的打电话功能以及之前已存在的其他功能。我们运用了专业的app测试工具，如Espresso、UI Automator等，对应用的性能、稳定性、兼容性等方面进行了深入的测试。

在测试过程中，我们发现了几个潜在的bug和性能问题，并立即进行了修复和优化。通过多轮次的测试和Debug，我们确保了项目的整体质量达到预期标准。

## 工作问题及重点关注

问题：在测试过程中，我们发现打电话功能在某些安卓版本上存在兼容性问题，导致部分用户无法正常使用该功能。

**重点关注**：针对上述问题，我们将重点关注以下两个方面：一是深入分析兼容性问题产生的具体原因，包括但不限于安卓系统版本差异、第三方库支持情况等；二是制定具体的解决方案，包括修改代码逻辑、升级第三方库版本或寻求厂商支持等，以确保打电话功能在所有目标安卓版本上都能正常运行。

## 工作分析与收获

**技术挑战与解决方案：**

在开发打电话功能时，我们遇到了如何确保通话质量和稳定性的技术挑战。通过调研和比较多种第三方通话服务SDK，我们选择了性能优异、稳定性高的服务商进行合作，并进行了细致的接口对接和测试工作。这一过程中，我们积累了宝贵的经验，提升了团队在集成第三方服务方面的能力**。**

**测试与Debug的重要性**：

通过全面的测试和Debug工作，我们及时发现并修复了项目中存在的问题，确保了项目的整体质量。这让我们深刻认识到测试与Debug在软件开发过程中的重要性。未来，我们将继续加强测试工作，确保每个功能模块都能达到高标准的质量要求。