第5章

# 提出并评价界面设计

界面设计是理解用户需求最为重要的手段之一。通过界面设计可以反思我们是否深入理解了用户的需求：如果无法设计出界面，那么我们一定还没有理解用户的需求。用户也可以利用界面设计来判断开发者是否正确地理解了其需求，同时反思曾经表述的需求是否真的是其所需要的：很多时候，用户并不真正地了解自己需要什么，他们需要看到软件的样子，甚至开始使用软件，才能搞清楚自己需要什么。最后，界面设计还能确保项目团队的所有成员都正确地理解了用户的需求，有效避免不同成员对于同一概念的理解出现偏差。

下面学习一些界面设计工具，并了解如何借助这些工具向用户介绍设计，另外，本章还会介绍如何评价界面设计。

## 5.1 绘制界面设计

首先从绘制界面设计开始。如果读者学习过用户交互设计类的课程，就一定听说过很多种现代化的界面设计工具，例如墨刀[①]、Axure[②]等。然而，在平时的工作中，绘制界面设计总是从一些有着几千年历史的工具开始的。没错，就是纸和笔（见图 5-1）。

图 5-1 使用纸和笔绘制界面设计

① https://modao.cc。

② https://www.axure.com/。

使用纸和笔绘制界面设计有着数不清的好处。它们轻便、成本低廉且随处可得，让你不必纠结于设备和网络连接的细节。它们即刻可用，没有任何加载时间，让使用者不会错过任何一点灵感。它们定位准确且性能高度稳定，关键时刻绝不拖后腿。它们显示清晰，可视角度大，并且没有任何学习成本，极大提升了多人协作的效率。这些特质让纸和笔成为人们整理思路、记录灵感，以及和团队成员们开展快速讨论时必备的工具。

当然，纸和笔的缺点也很多。绘制在纸上的界面设计不太方便归档和传递，容易丢失，并且修改起来非常麻烦。当开始遇到这些问题时，就可以使用软件来绘制界面设计了。通用的绘图软件，例如 LibreOffice Draw、Microsoft Visio 等，可利用一些简单的黑白线条来绘制界面设计，如图 5-2 所示。

如图 5-2 所示的线框图在很多情况下已经足够使用者完成小型项目的界面设计了。为了使界面更加丰富美观，可以使用更加专业的界面设计工具，如墨刀、Axure 等。图 5-3 是使用这些工具将图 5-2 的线框图重新实现的效果。可以看到，对于比较简单的界面设计来讲，使用专业的界面设计工具未必会取得更好的效果。事实上，设计的效率也未必获得显著的提升。因此，有必要根据项目的复杂度来选择合适的界面设计工具。

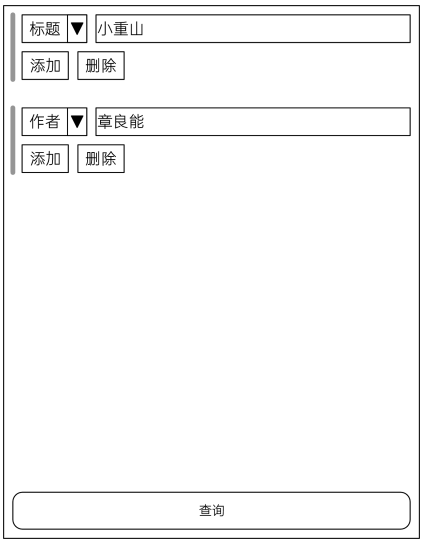


图 5-2　诗词搜索页原型设计（一）


图 5-3　诗词搜索页原型设计（二）

根据 3.5 节的观察与访谈总结，此外使用线框图绘制了界面设计。为了避免这些界面设计占用太多的篇幅，下面将它们以一个单独 PDF 文件的形式提供。

界面设计文件

在我们的设计中，诗词收藏页用于帮助用户记录已经背过的诗词，诗词详情页则用于呈现诗词的内容。利用诗词收藏页与诗词详情页，就可以让用户（奶爸 F）帮助宝宝 B 复习诗词了。用户可以使用诗词搜索页搜索诗词，并在诗词详情页中收藏诗词，从而将新的诗词添加到诗词收藏中。应用内建有一个诗词数据库，并支持搜索诗词的题目、作者，以及内

容片段,方便用户收藏诗词。用户还可以利用数据同步页在不同的设备之间同步数据。最后,应用启动时会显示一首推荐的诗词,并显示出一幅优美的背景图片,从而为用户带来优雅的使用体验。本书中将这个应用称为每日诗词 X(DailyPoetryX,简称 Dpx),其中 X 代表 Xamarin。

当然,我们给出的设计只能满足用户最为基本的需求。由于缺乏记录复习进度、自动规划复习方案等功能,目前的设计难以称得上一个完美的设计。不过,给出这种设计是为了优化后续的学习过程。基于目前的设计,我们可以学习到足够多样的知识、技术和技能,同时又能避免大量的重复性内容。事实上,实现上文提到的功能与实现诗词收藏功能的方法是差不多的。为了避免大量的重复性内容,从而提升学习效率,本书省去了这些功能。

## 5.2 形成操作动线

界面设计图只能描述软件的静态状态。仅仅依靠界面设计图,用户和团队成员可能不容易理解软件的动态行为,也就是软件是如何与用户交互的。为了让用户和团队成员理解软件的动态行为,还需要基于界面设计形成操作动线。

操作动线是一个不容易解释,但人"一看就懂"的概念。图 5-4 就是"今日推荐"页的操作动线。用户在"今日推荐"页中点击"查看详情"按钮就会跳转到推荐详情页。在推荐详情页中点击"在本地数据库中查找"按钮则会跳转到诗词搜索页,同时将诗词的题目和作者作为查询条件。利用操作动线,可以很容易地描述软件的动态行为。

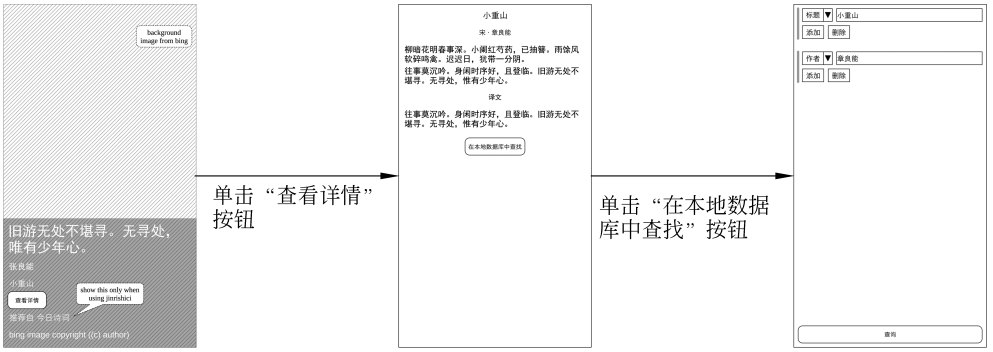


图 5-4 今日推荐页操作动线

下面的视频同样将 Dpx 应用的操作动线绘制了出来。

操作动线除了可以静态呈现之外,还可以动态呈现。动态呈现操作动线的方法。

Dpx 应用操作动线

动态呈现操作动线

这种动态呈现的操作动线通常是非常有趣的，也能够帮助开发者快速地了解其设计是否存在不合理之处。当用户或团队成员不知道应该如何进行下一步操作时，该设计可能就已经存在问题了。

除了采用上面的办法，还可以采用专业的界面设计工具将操作动线制作为可交互的软件原型。这些工具的教程和文档中能够找到这方面内容，本书中就不做过多介绍了。

## 5.3　评价界面设计

设计软件界面很重要的一环是评价设计是否合理。在评价界面设计时，最为著名的方法可能就是 Jakob Nielsen 提出的 10 条启发式规则。它们被称为“启发式规则”，是由于这些规则并不是严格的、可量化的标准，而是一种指导性的、方向性的准则。这些规则包括[①]：

- 可视性原则(Visibility of system status)：软件应该在合适的时间内，以合适的方式告诉用户软件当前的状态。例如，在“今日推荐”页加载数据时，Dpx 应用会提示“正在载入”。
- 不要脱离现实(Match between system and the real world)：软件应使用用户熟悉的语言描述问题，并以符合常理的方式与用户交互。例如，很多人搞不清楚什么是“借记卡”“贷记卡”。因此，使用“储蓄卡”“信用卡”的称呼可能更方便用户理解我们在说什么。
- 用户拥有自由控制权(User control and freedom)：用户可能会由于误操作而进入不想要的界面。此时，用户应该可以轻松且安全地从界面中退出，同时还应该支持撤销和重做。例如，软件的设置界面总应该提供一个关闭按钮，使用户能够在不保存设置的情况下关闭设置界面。
- 一致性与标准化(Consistency and standards)：使用统一的术语与操作模式，避免用户猜测不同的术语或操作是否指的是同一种功能。同时，软件应该遵循开发平台的交互规范。例如在绝大多数情况下，Ctrl＋C 都应该是复制，而不是其他功能。
- 预防出现错误(Error prevention)：通过良好的交互设计来避免用户做出错误的操作，或至少应该在用户执行可能发生错误的操作前给出提示。例如，文本编辑器可以在用户关闭文件时自动保存文件，或者提示用户是否需要保存文件。
- 易于认知，免于记忆的操作(Recognition rather than recall)：不应该要求用户记住如何操作软件，而应该让用户能够很容易地判断软件的功能或获取帮助。作为一个反例，试着在任何一个网络银行软件中查看你的储蓄卡的完整账号。
- 使用体验灵活高效(Flexibility and efficiency of use)：为熟练的用户提供加快操作的方法，在确保在新手用户正常使用软件的同时，提升熟练用户的操作效率。集成开发环境(IDE)的各种快捷键和快捷代码生成工具就是很好的例子。

① https://www.nngroup.com/articles/ten-usability-heuristics/。

- 优雅且克制的设计(Aesthetic and minimalist design):避免向用户提供不相关或很少派上用场的信息,确保用户能够将注意力集中在有价值的信息上。作为一类反例,用户很少真正了解其在操作各种软件时都同意过哪些条款。
- 帮助用户恢复错误(Help users recognize, diagnose, and recover from errors):以用户能够读懂的方式提示错误,明确地阐述错误,并提供可能的解决方案。想实现这一点是很困难的。以 Windows 系统的"疑难解答"功能为例,尽管它能够解决的问题越来越多,但鲜有用户记得上一次它成功地解决用户的问题是在什么时候。
- 帮助和文档(Help and documentation):尽管最理想的情况是用户不需要文档也能使用软件,开发者仍应为用户提供便于访问和搜索的文档。这些文档应该专注用户要完成的任务,提供详细的步骤,并且避免写得过分冗长。

这些启发式规则有助于开发者发现界面设计中存在的问题。需要注意的是,很多时候,即便有这些启发式规则的帮助,开发者也很难发现自己设计中的问题。此时,邀请团队成员甚至外部人员来协助评价测试会是一种非常有用的方法。可以将这些启发式规则与操作动线结合起来,并运用 3.3 节讨论过的访谈方法从其他人处获得反馈。这种组合式的方法会更有助于设计的评价与改进。

## 5.4 动手做

经过前面的学习,你应已选定了项目问题,并获得了用户的需求。应用这一章讨论过的方法,给出原型设计和操作动线,并与团队成员以及用户沟通,了解是否已经互相理解,并且针对需求达成了一致。你们几乎一定会发现一些彼此之间存在误解的情况。总结发生这种误解的原因,并讨论如何避免类似的误解发生。