

02 无规矩不成方圆

——设计规范的建设

文 / 周建波、朱兰民

第 1 节 规矩成就方圆

孟子曰：离娄之明，公输子之巧，不以规矩，不能成方圆。——《孟子》

设计规范是用户体验的最低标准！——吴明

支付宝作为一个大型的“生活服务类平台”，既有官方自营应用，也有第三方接入应用，数十个应用共计数百个页面，并且还在不断发展壮大。我们将支付宝定义为一个系统级的应用一点儿都不夸张，其复杂度已经逼近一个完整的“生态系统”。

在这样一个系统级的应用中，我们面临的设计挑战相当复杂：如何降低用户在复杂系统中的学习和认知成本？如何让用户在所有页面都有良好的体验？如何让用户在所有页面中获得一致的品牌认知？

为应对系统级的设计挑战，我们必须建立系统级的设计规范。你

可能会问：什么叫“系统级的规范”？其实一开始我们也没有“系统级的规范”这样的概念。但经过长时间的实践，我们发现简单的设计规范并不能很好地应对支付宝这样复杂的产品。于是，我们不断对设计规范进行补充和完善，最后自上而下地建立了一套设计规范系统。

1. 规范 1.0

我们最早的设计规范只是各种控件的视觉设计源文件。这样的规范保证了页面基础元素以及框架的一致。但是，这些视觉源文件并没有定义任何的交互规则，实际工作中还是会出现各种不一样的产品体验。例如，某些页面的弹框出现方式不一样；什么时候应该用小气泡，什么时候该用弹框。规范 1.0——样式库如图 1 所示。



图 1 规范 1.0——样式库

2. 规范 2.0

我们的产品越来越复杂和庞大，用户面临的学习和认知成本成倍上升。我们必须对页面中通用交互规则进行定义，打造一致的操作体验，降低用户的学习成本。为了最大限度地降低用户学习成本，我们最大限度地复用系统原生的交互规则。这套规范进一步保障了重要元素的交互行为和视觉样式的一致性。但是，当遇到一个规范里没有的东西时要怎样设计，设计师才能既可以发挥自己的创意，又不会和整体的体验风格相背离？交互规范文档如图 2 所示，视觉规范文档如图 3 所示。

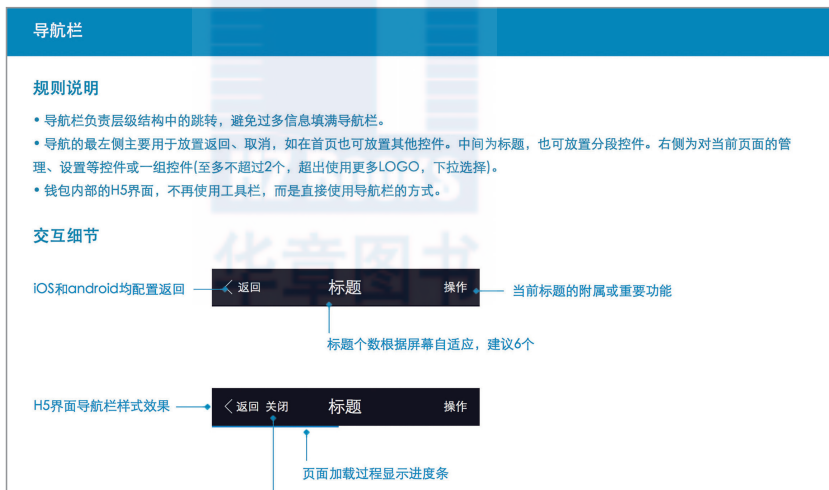


图 2 交互规范文档

3. 规范 3.0

交互规范和视觉规范都是死的，设计是活的。要想进一步提升内部的设计一致性和整体的设计质量，团队要有一致的设计价值观。大家都

知道什么样的设计是好的，什么样的设计是不好的。一个代表团队设计价值观的东西——设计原则应运而生。至此，我们自上而下建立了一套设计规范体系：设计原则、基础规范、单元控件、标准组件，然后落地到开发控件库，如图 4 所示。从设计思维到最后实现的每个环节，对设计的一致性和可用性进行整体把控。

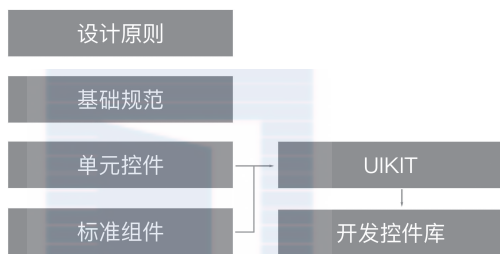


图 4 规范 3.0 自上而下的设计规划体系

第 2 节 设计原则

设计原则是做设计时必须遵循的标准，可用于指导设计和衡量设计方案的优劣。我们要什么样的设计，我们不能接受什么样的设计，设计原则让团队成员形成一致的认知。

大家可能都看过很多类似于“移动设计 N 原则”“可用性原则”之类的教程、文章，但是这些设计原则都是别人通过自己的经验总结出来的，是适合别人的原则。如果你直接照搬过来当作真理，那就不合适了。所以，我们在建立自己的设计原则的时候，应该结合自己产品的特点和实际的案例，制定适合自身的设计指导原则。

1. 简单的力量

2013年6月13日，支付宝正式推出“余额宝”。它凭借“傻瓜式”的理财方式，让没有任何理财专业知识和技能的人都能理财。以此在中国掀起了一股互联网金融狂潮。余额宝宣传广告如图5所示。



图5 余额宝宣传广告

余额宝的成功让我们看到简单对于支付宝来说多么重要。简单不仅能提升用户体验，还能实实在在地创造价值。因为简单，更多的用户接受你的产品，并且使用你的产品；因为简单，你的转化率也会更高，会产生更多的价值。

像余额宝、微信这些简单的产品，对市场产生了深远的影响。它们简单易用，所以被大众接受；它们值得信赖，所以赢得用户。现在智能设备普及率越来越广，上自几十岁的大爷大妈，下至几岁的小孩，都已经成为App的用户群。所以，我们的简单原则不仅能使界面看起来简洁，还能使交互流程简化，使整个产品轻量化。

为了让产品能够更简单，我们在设计上制定了以下两个原则：

- 一个页面只做一件事情，突出两个重点。
- 删除不必要的内容，隐藏次要的内容。

(1) 一个页面一件事情

与 PC 相比，移动终端的屏幕小了很多。App 的一个页面能展示的信息本来就非常有限，不可能像 PC 一样堆满各种不同的信息。况且，App 的使用环境还非常不稳定，如走路、坐车、单手、双手等，这些进一步限制了一个 App 页面只能做一件事情。

支付宝里很多产品都是以任务为导向的，帮助用户实现某个确定的任务目标，如转账、缴费等，如图 6 所示。在任务导向类的页面中，这个原则显得尤为重要，因为我们希望用户可以专注且快速完成当前任务。

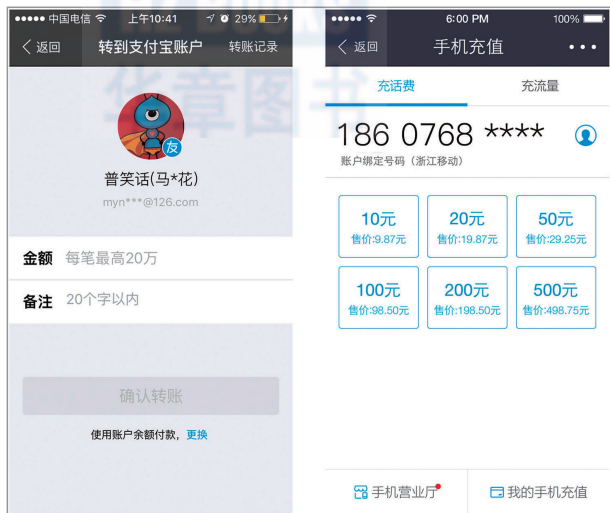


图 6 一个页面一件事情

(2) 删除 & 隐藏

人在处理信息、学习规程和记忆细节方面的能力是有限的。现实中，人可能还面临各种中断和打扰，这些都进一步限制了人的能力。界面中过多的小细节会增加用户的认知负担，就像路障会降低人们的行走速度一样，过多的小细节会降低用户的使用效率。

删除那些可有可无的功能、多余的选项、冗余的文字、花哨的修饰，可以减轻用户的负担，让用户专心做自己想做的事。界面清清爽爽、简简单单，不去分散用户的注意力。

1) 删除

我们带着“删除不必要功能”的目的，重新来审视支付宝的生活缴费，看看哪些不必要的功能可以删除。

- 确定主要任务和次要任务，并排定优先次序。专注解决用户优先级最高的需求，然后再考虑满足用户其他目标。
用户使用生活缴费，主要目的就是完成缴费；然后是新增缴费；最后是辅助功能——账户分组。
- 要知道能够满足主流用户的“足够好”的生活缴费与“完美”生活缴费有什么区别。如 9.2 版的支付宝生活缴费（图 7 中左图），要求补全家庭地址、芝麻信用、家庭成员，这么多的附加元素，在绝大多数情况下，没人使用。这些功能就是不必要的，可以删除，如图 7 中的右图所示。



图7 删除不必要功能

2) 隐藏

有些功能用户很少使用，但又是必需的。例如，自定义、删除、设置等，隐藏这些不常用但不能少的功能（因为可以少的功能我们已经删除了），可以为移动界面节约很多空间，让界面简单。用户也不会因为它们而分散过多的注意力，更聚焦主要任务，如图8所示。

隐藏的功能在用户需要的时候会出现在合适的位置。例如支付宝转账到卡，默认隐藏了到账时间，以简化页面。当用户填好卡号等信息的时候会自动出现到账时间，如图9所示。



图 8 隐藏不常用功能



图 9 适时出现

2. 高效的执行

生活的节奏越来越快，高效是一款产品必备的素质。减少等待、稳

定快捷，才能留住用户。外面的世界很精彩，用户不想等待，用户不想填写没完没了的表单，用户不想总要跳转才能看到想看的内容，用户不想不停地点击……

关于高效的问题，我们应该着眼全局，并且思考“怎么样让我们的应用更快、更好用”。所以高效问题不仅仅是关于设计的问题。有跟技术和性能相关的时间问题，“1秒钟等待”可以让你的应用迅捷如奔豹；有跟行为流程设计相关的点击问题，“1次点击”可以让你的应用飞起来；有跟移动设备输入相关的人机工程问题，“1根手指”可以让你的应用连小孩都能使用；还有可能跟信息结构相关的产品问题，“潘神的迷宫”带你的用户走出页面的迷障。

(1) 1秒钟等待

不知道大家有没有这样的经历：风和日丽的下午，慵懒的阳光洒在身上，你漫步在一条商业街上。闲逛中发现一个店铺在搞活动，于是掏出手机扫描二维码，无奈网速太慢，手机一直在安全扫描，你只好耐心地等。漫长的等待过后，终于打开一个页面，但那是空白页面！而且，这个页面还没有任何提示。为了5折的优惠，你决定再等等。1秒、2秒、3秒，终于页面再次跳转，进到一个全新的页面，而且显示了加载的进度。5秒、6秒、7秒，页面终于刷出了大部分内容，无奈活动按钮还是犹抱琵琶半遮面，不肯出现。于是你一遍一遍地刷新，loading，刷新，loading……

许多研究都表明，用户能够忍受的等待时间为6~8秒，如图10所示。也就是说，8秒是一个临界值，如果你的页面打开速度过慢，等

待 8 秒以上，大部分用户会离你而去。

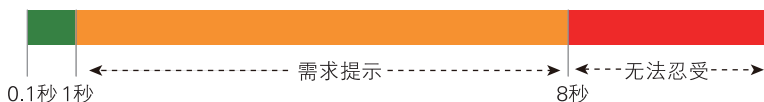


图 10 用户等待时长

在 0.1 秒内显示反馈结果用户是可以接受的。

1 秒是用户保持不间断的思维流的限定时间，用户会注意到这样的延迟。

8 秒是保持用户关注当前界面的极限时间，超出 8 秒用户会愤然离场。

没人愿意等待。“1 秒钟等待”是一种比喻，表示用户对速度和高效的期盼与要求。其实，等待的时间主要取决于服务器、设备的性能以及网络的速度。用户是否需要等待主要取决于 RPC 的策略。

RPC (Remote Procedure Call Protocol) 是指手机客户端需要调用网络服务端的资源时，进行的远程数据调用过程。我们用“RPC”泛指客户端需要和服务端发生数据交换的场景。

虽然设计师不能决定服务器和设备的性能，也不能提升网速，但是我们可以优化 RPC 策略，减少等待的次数，也可以优化加载效果，减少用户的等待时间。虽然某种程度上这只是“障眼法”，但却能够获得用户的认同，改善用户的体验。

除去服务器、设备性能、网速这些不可控因素，用户的等待主要由一次次 RPC 请求造成。如果我们在做设计的时候考虑到 RPC 请求的逻辑和策略，并且适当地优化，做到用户的一次操作只触发一次 RPC 请求，尽量减少 RPC 次数，就可以减少用户的等待次数。

一次用户操作不允许有多次 RPC（指和这条操作主流程相关）。

以用户使用支付宝扫描二维码的体验为例，用户要经历 3 次 RPC 请求才能正式进入页面加载，如图 11 所示。



图 11 扫码 RPC 策略及页面流程

从图 11 中我们可以看到一个简简单单的扫码，用户经历了 3 次等待、2 次跳转，真是千辛万苦啊！我们可以优化一下这个 RPC 请求的逻辑，合并识别二维码和打开哪个页面的 RPC 请求，如图 12 所示。



图 12 优化的 RPC 策略及页面流程

在图 12 中，我们看到这样的 RPC 策略让用户少等待了一次，少了一次页面的跳转，给用户的体验就比之前要高效一些。

(2) 转移注意力

转移注意力是减轻等待的负面影响的常用手段。其实，在现实生活中，我们常常使用转移注意力这一方法。比如，一些服务较好的饭店，在客人排队等待就餐的时候提供各种小吃和休闲游戏。客人有好吃的、好玩的就会忘记等待的烦恼。这种方式在应用的设计中也同样管用。

支付宝的明星产品“余额宝”，当用户进入页面的时候会从服务端获取金额信息。在这一过程中伴随着金额数字随机变换以吸引用户

注意，使用户不会有等待之苦。如图 13 所示。

(3) 一次点击

懒，是人的天性。交互设计师的天职就是将用户从繁重的交互操作中拯救出来，提升产品的可用性，让用户方便快捷地完成任务。在交互设计中，应尽可能地减少额外的点击，做到一键完成任务。一次点击意在减少用户操作次数，提高操作效率，成全人类“懒”的天性。



交互产品经常包括一些不必要的额外点击，对于用户而言，这些不必要的操作都是附加工作。附加工作消耗用户的精力，又不能直接实现用户的目标。消除附加工作，可以提升操作效率，改善产品的可用性。交互设计师只有对产品中附加工作高度敏感，才能把产品设计得更高效。

图 13 余额宝页面加载动态

支付宝 9.2 版本以前的手机充值从选择金额到付款需要 4 次点击：
①点击金额唤起选择 picker；②滑动选择金额；③点击完成关闭 picker；
④点击“立即充值”进入付款页面，如图 14 所示。

9.2 版本改版以后将充值金额平铺展现在用户面前，用户只需要一次点击选择充值金额即可进入付款页面，如图 15 所示。



图 14 无尽的点击



图 15 手机充值交互优化

(4) 三级跳

在 PC 网络时代，Web 页面间反复跳转是再正常不过的了。从搜索页面到目标网站首页，首页再到详情，详情页再到推荐页面，推荐页再回到首页……但是，你敢在手机应用上试一试吗？这么混乱的跳转关系，用户可能真的就进入了你为他设置的页面迷宫，不知道自己在什么地方，不知道返回到哪个页面，不知道怎么快速回到入口。

我们来看图 16 所示的某热门应用的页面跳转案例：首页→详情页→个人页→详情页→个人页……可以无限制地深入和循环。当用户回过头来想要返回的时候就迷糊了。左上角的返回是到哪儿的？如果是返回上一个页面，那回到首页要点击多少下？

在 PC 网页上因为有顶部全局导航和面包屑导航，这些跳转可能都不是问题。但是在手机屏幕上是没有这些导航的，只有左上角的返回，这就完全交代不清楚了。所以，在手机应用页面之间的跳转不能太多。



图 16 某热门应用

标签栏、抽屉式、宫格式、列表式、轮播式、唤起式，移动设计里的导航模式就这几种，都是比较简单的结构和模式。如果在移动设计里面设计了太深的信息架构，一会让用户迷惑，二会大大降低操作效率。所以，移动设计中常用的内容应该在 3 个层级以内看到。

手机上能不跳转就不跳转，下面的案例可以帮助我们减少页面的跳转，将用户带出“潘神的迷宫”。



图 17 模态窗口

模态窗口就是对当前页面的内容进行操作，用户不用离开当前页面，从而减少了页面的跳转。我们常见的模态对话框，常用来报错或者提醒用户。如图 17 所示。

3. 人性的对话

用户使用我们的应用过程其实就是一个人机对话的过程。界面中展示的信息是系统向用户传递信息，用户的操作是在向系统传递信息，这一来一往就是对话。打造一个有人性的 App，营造和谐的人机对话，是

我们提出人性化原则的目的。我们希望用户可以把我们当成生活中一个可以对话的助手。

(1) 适时反馈

人与人的交流中，我们无法忍受的一种情况是：对方对自己说的话没有反应，对自己的行动视而不见。这是一种极其不尊重人的行为，我们会在内心鄙视这种人，甚至抓狂！及时恰当的反馈能告诉用户下一步该做什么，帮助用户做出判断和决定，让用户知道系统运行良好稳定。所以要营造和谐的人机对话环境，我们必须做到适时地反馈。下面定义了几个最基本的原则，来保障支付宝对用户做到适时反馈：

- 为用户的操作提供必要、积极、即时的反馈。
- 根据内容的重要程度选择合适的反馈形式。
- 避免过渡反馈，以免给用户带来不必要的打扰。
- 不要打断用户的意识流，避免遮挡用户可能回去查看或操作的对象。

1) 反馈方式

移动界面设计中反馈的方式大致有 5 种：对话框、小气泡、多态按钮、动画、声音或震动，如图 18 所示。所有的提示都应该在恰当的时候出现在恰当的位置，为用户提供必要、积极、即时的反馈。



图 18 反馈方式

- **对话框**：对话框带有一两句说明文字和两个操作按钮，用于确认和取消重要操作（比如，是否删除内容），通常会用明显的颜色，突出显示可能造成损失的操作项（比如，“删除”“不保存”）。对话框的出现，强迫用户关注弹窗内容和操作，并屏蔽背景的所有内容不可操作，是对用户打扰最大的反馈提示，也是最强的反馈方式。通常用户都想赶快关掉对话框，以便接着完成被打断的操作。所以，对话框中的文案要尽量言简意赅，帮助用户快速了解和做出决策，如图 19 所示。



图 19 对话框

- **小气泡**：气泡也可以叫“Toast”，这其实是一种弱化版的对话框。它就像气泡一样，在界面上展示短暂的时间（5秒以下），然后自动消失。它不强制用户做任何操作和确认，所以对用户的打扰比对话框小很多。气泡一般用来确认用户执行的任务状态或者操作结果，如图 20 所示。

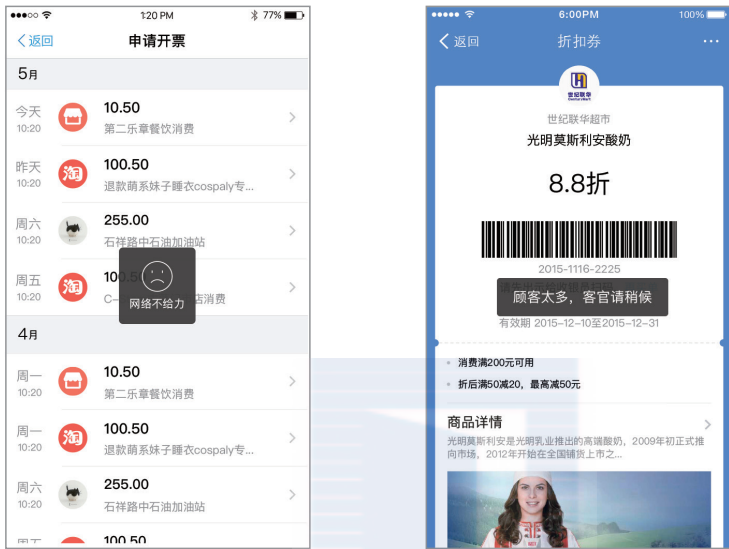


图 20 小气泡

- **多态按钮**：按钮是虚拟世界隐喻现实世界的产物，是对现实世界的按钮和开关的模拟。而现实世界中的按钮会对用户的操作提供实实在在的物理反馈。所以，为了符合用户的心智模式，界面中的按钮也应该为用户的操作提供反馈，否则用户不知道发生了什么。当用户在屏幕上按下一个按钮或链接时，也需要有状态的变化，让用户知道界面已经接收到他的操作了，如图 21 所示。
- **动画**：动画能给用户提供有意义的反馈，帮助用户直观了解操作的结果。并且不打扰用户的操作，用户体验更流畅。而且精美有趣的动画，能给用户留下深刻印象，提升使用时的愉悦感。

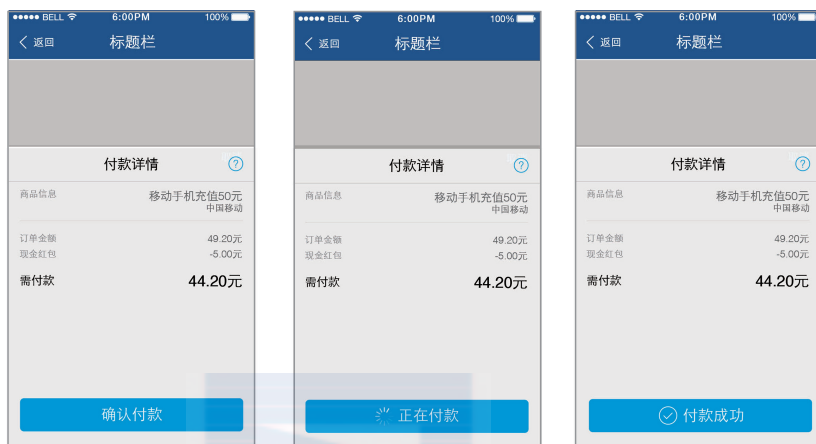


图 21 多态按钮

手机淘宝将商品收入购物车时的动画，就是一种比较优雅的反馈方式，如图 22 所示。

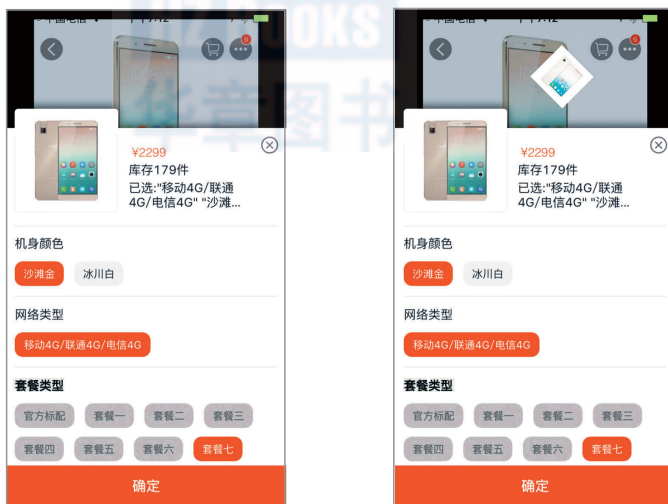


图 22 动画

- **声音或震动**：前面讲到的反馈方式都是虚拟界面模拟出来的，而声音或震动能带给用户更真实的物理反馈。声音或震动的反馈给用户的感觉也应该是最真实和自然的。例如，手机虚拟键盘按下时的“啪啪”声，短信、邮件发送成功后“嗖”的一声，拍照时的“咔嚓”声，都是我们设计中在声音方面反馈比较好的。

恰当的声音反馈有画龙点睛的效果，但过多地使用可能会变成一种打扰。所以，我们不能将声音作为主要反馈，并且要给用户关闭提示音的权利（因为用户所处的环境多样，可能很吵而听不见声音，也可能不适合打开声音）。

震动是一种更强烈的触觉反馈，可以用来加强声音的反馈。

2) 过度反馈

对话框和小气泡的视觉形式是悬浮在当前界面之上的，用户可以理解对话框与当前界面之间的关系，完成对话框的操作之后可以继续原有操作。它的本意是减少页面的跳转，尽量让用户沉浸体验。但是有的设计师会滥用对话框，反而打扰了用户的体验流程，成为万恶的对话框。试想一下，如果每想说一句话之前，你都要在脑子里面弹出一个对话框问自己“你是否真的要说这句话”，这是一种多么糟糕的体验。下面给大家举几个反例。

支付宝 9.2 版本之前，用户主动关闭收银台，会弹出对话框让用户确认是否退出，其实显得十分多余，如图 23 所示。



图 23 过度反馈 1

支付宝 9.0 版本以前的服务窗，在删除服务窗的时候会展示删除成功的小气泡，如图 24 所示。其实完全可以用现在的删除动画代替。



图 24 过度反馈 2

(2) 情感关怀

情感是人对客观事物是否满足自己的需要而产生的态度体验。——《心理学大辞典》

人性和情感是紧密相连、不可分割的，我们在思考人性化的时候，一定不能忘了用户的情感需求。用户的需求和期望得到满足时会产生愉快、喜爱的情感；反之，就会苦恼、厌恶。所以，当用户在支付宝里受挫的时候，我们要及时地给予情感上的安抚和关怀；当用户在支付宝里完成任务的时候，我们要及时地给予情感上的肯定和强化。

1) 安抚和关怀

等待、报错、系统繁忙等场景都会使用户产生挫败感。长时间的等待会让用户感到烦躁；犯错误让用户感到苦恼；系统繁忙甚至让用户感到厌恶。如果我们能巧妙地处理好这些场景，降低用户的挫败感，就能体现设计在产品中的价值。

例如，支付宝在“双11”大促以及发春节红包的时候，用户可能会遇到系统繁忙的情况。但是这个时候却是用户最着急付款的时候，因为大家都在抢货啊！这个时候我们怎么能通过情感的安抚降低用户的挫败感呢？我们赋予系统人格特性，用通俗易懂的语言和虔诚的口吻，跟用户展开“人与人”之间的对话，如图 25 所示。

2) 正向强化

在用户完成某个任务或者其他正向场景的时候，我们要强化这种正向的情感，加深用户的情感体验，如图 26 所示。



图 25 安抚和关怀



图 26 正向情感强化

图 26 左图为用户进入余额宝页面时, 我们通过金额数字的增长动画, 加强用户在查看收益和本金时候的喜悦感, 加深用户对余额宝这个产品乃至支付宝的正向情感。

图 26 右图为用户开启支付宝会员的时候, 我们通过一个完整的页面和动画告知用户获得的排名与会员等级, 强化用户获得会员身份和等

级时的自我认同感。

(3) 智能服务

智能手机要够智能。充分利用设备的硬件特性、系统的数据，与设计进行完美的融合，让用户感受到我们是懂他的，是聪明的。手机的传感器赋予我们对外界的感知能力，海量的用户特征数据赋予我们渊博的知识，强大的后台计算服务器赋予我们超强的大脑，我们的 App 完全可以像人一样去思考，思考如何为用户提供更好的服务，如图 27 所示。

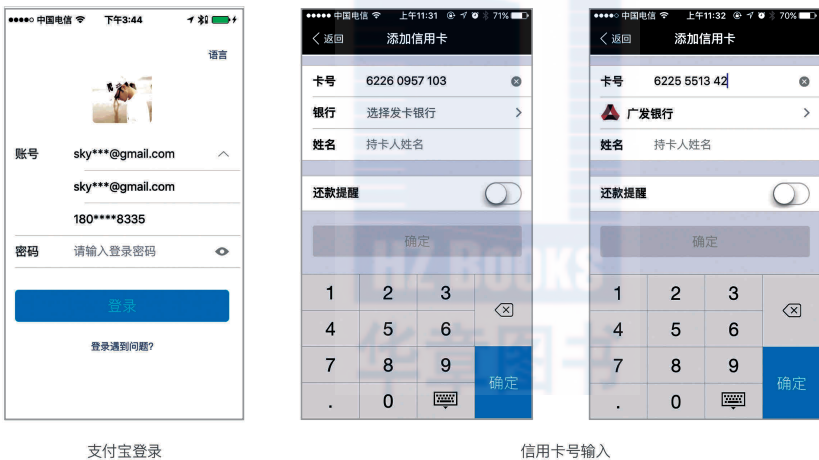


图 27 智能服务

图 27 左图为支付宝登录，用户在应用中填写过的信息都可以记录在系统中，免去用户二次输入的麻烦，提升用户的操作效率。

图 27 右图为新卡号输入，自动将卡号分隔为 4 个数字一段，方便用户阅读和核对，并且根据卡号自动识别所属银行，减少用户的输入操作。

第3节 基础规范

设计原则在思维层面指导我们做出符合整体体验价值观的设计决策。根据简单、高效、人性化的设计原则，我们要对页面的基本组成元素进行一致性的定义和规范化的建设。文字、图片、线条、颜色是构成页面的最基本元素，统一这些基本元素对于整个产品的视觉风格和品牌传达的一致性起着至关重要的作用。它们就像建筑的基本材料，决定建筑最终的风格和特点，正如用木材建造中国及东方建筑，用石材建造西方建筑。

1. 设计风格

设计风格是整合品牌传播的一种表现，它能理性地反映品牌个性与共性，从而建立品牌知名度、美誉度、忠诚度及品牌联想度。

支付宝给用户传达简单、高效、人性化的理念。而极简主义（Minimalism）指设计从功能出发，形式以满足功能而存在，没有功能性的修饰全部去掉。这种设计风格，感官上简约整洁，品位和思想上更为优雅。

极简主义设计时，只有图片、颜色、字体及大小、线条、icon 风格、留白的处理等设计元素之间完美配合，才能达到理想的效果。

2. 色彩

品牌色在界面设计中的使用应同时具备品牌识别性以及界面设计功能性，色彩的运用应达到信息传递、动作指引、交互反馈，或是强化和

凸显某一个元素的目的。支付宝品牌色如图 28 所示。

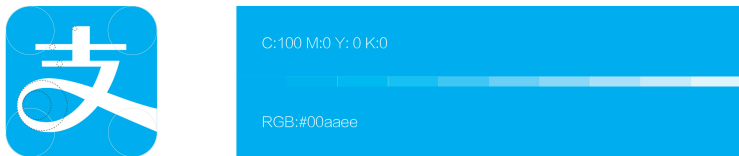


图 28 支付宝品牌色

人们的大脑记忆过程主要分为 3 个阶段：识记、保存、重现。在实际设计中，我们可以利用这一特点，更好地传达品牌的视觉感知。支付宝移动端通过视觉感官来增强品牌色的引用主要有 3 种途径，如图 29 所示。

- 当前选中项（图 29 中的左图）；
- 主操作按钮（图 29 中的中图）；
- 操作类文案信息（图 29 中的右图）。

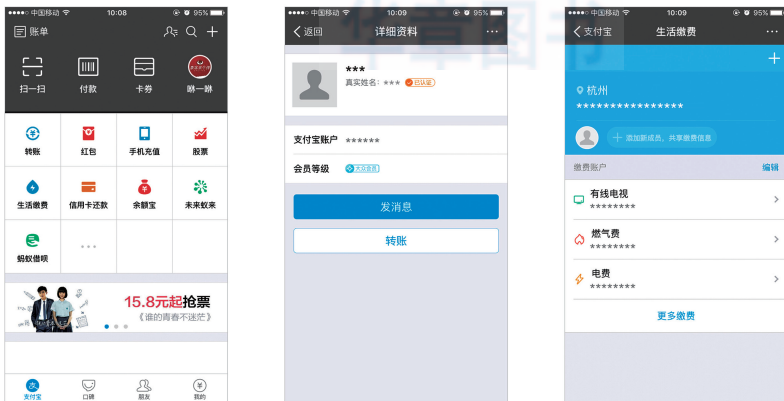


图 29 品牌色引用

3. 字体

很多设计师在产出 App 时，都会遇到字体选择的问题，手机系统的不同，所使用的字体也是不同的。有些个性的 App，若字体也做个性设计，会产生 3 个方面的影响：

- 增加了 App 的安装包的大小。
- 用户在使用该 App 时，增加额外的学习和适用成本。
- 个性字体与系统字体没有统一性。

建议大家采用系统字体做设计方案的产出。下面介绍一下大家经常接触的两大系统字体及属性。

(1) 字体集

1) iOS 系统

在 WWDC 2015 大会上，苹果公司专门在 OS X El Capitan 中为中国用户打造了全新字体——苹方字体，其字体具有现代感的外观、清晰易读的屏幕显示效果，并同时支持简体中文（PingFang SC）、繁体中文（PingFang TC）、香港中文（PingFang HK），看来苹果公司还是很注重中国人的用户体验。iOS 默认字体如图 30 所示。



图 30 iOS 默认字体

2) Android 系统

谷歌公司就没有这么人性化了，中文字体至今尚未更新，依然沿用 Droid Sans Fallback 字体，或许是字体库对手机厂商开放，可以自行定义。

自从 Ice Cream Sandwich 发布以来，Roboto 就是 Android 系统的默认英文及数字字体。宽度和圆度都轻微提高，从而提升了清晰度，并且看起来让人更加愉悦。Android 默认字体如图 31 所示。



图 31 Android 默认字体

(2) 行高

1) iOS 系统

可能很多设计师都遇到这样的问题。在设计过程中，采用默认的字行高产出设计稿，但在视觉验收阶段，却发现技术开发的页面与设计稿不一致。例如：文字与图片的间距、文字与文字间距、文字与边框间距等都不同。

解决此类问题其实很简单，文字行高采用与技术开发对等的参数产出设计稿即可，图 32 列举了一些常用的文字行高。

24px	文字20px: 支付宝 ABC abc 0123
28px	文字22px: 支付宝 ABC abc 0123
30px	文字24px: 支付宝 ABC abc 0123
32px	文字26px: 支付宝 ABC abc 0123
34px	文字28px: 支付宝 ABC abc 0123
36px	文字30px: 支付宝 ABC abc 0123
40px	文字32px: 支付宝 ABC abc 0123
42px	文字34px: 支付宝 ABC abc 0123
44px	文字36px: 支付宝 ABC abc 0123

图 32 常用的文字行高

从图 32 中可以总结出一个公式:

$$Y = 2 \times \text{ceil}(X/10) + X$$

式中: X 为字号 (sketch 里的字号); Y 为行高; $\text{ceil}()$ 为向上取整数。

此公式方便我们在产出设计稿时, 按照文字大小, 对应地设置文字行高, 便可以确保设计稿与技术开发的页面中的文字行高一致了。

2) Android 系统

Android 由于各机型的字体不同, 字体库对手机厂商的开放, 文字行高就很难找统一的规则。在实际产出设计稿时, 采用字体 Droid Sans Fallback、Roboto, 文字行高为系统默认, 不做设置, 如图 33 所示。

30px	文字30px：支付宝	40px	文字30px：ABC abc 0123
33px	文字33px：支付宝	43px	文字33px：ABC abc 0123
36px	文字36px：支付宝	46px	文字36px：ABC abc 0123
39px	文字39px：支付宝	49px	文字39px：ABC abc 0123
42px	文字42px：支付宝	52px	文字42px：ABC abc 0123
45px	文字45px：支付宝	55px	文字45px：ABC abc 0123
48px	文字48px：支付宝	58px	文字48px：ABC abc 0123

图 33 中文、英文、数字行高

(3) 标点

中文字在国际 GB 2312—1980（汉字国际码）中采用全角字符，输入中文时，所使用标点同时应采用全角字符。这样就与中文字占的字节数保持一致，展示出来的中文字与标点所占的间距保持统一。

同样地，英文、数字、特殊字符采用半角输入模式，其内容所使用的标点也采用半角标点。

(4) 空格

“空格”估计大家并不陌生，它是键盘里最大的一个按键。如何把“空格”应用到实际的设计中，总结以下 3 点经验：

- 中英文混排时中文与英文之间加半角空格，方便用户在阅读时进行区分。
- 数字与单位之间需要增加空格，方便用户精确查阅数字（支付宝用户对数字非常关注，数字等同于金额）。但度、百分比与数字之间无须增加空格。

- 中文之间链接文字需增加空格。在技术上实现链接时，可增大可点区域，同时给用户传递可点击感。

(5) 对齐

人们的阅读习惯一般是从左到右，人的两眼是横着排列的，眼睛视线横看比竖看要宽，横看时眼和头部转动较小，自然省力，不易疲劳。

在实际的文字排版中，中文或英文均可以采用左对齐的方式，如图 34 所示。

支付宝 App

支付宝内置风靡全国的平民理财神器“余额宝”，还有还信用卡、转账、充话费、缴水电煤全部免费，而且还提供了以你为中心的餐饮、生活服务、理财保险、场景关系等一系列全新服务

图 34 文字对齐

数字通常采用右对齐或小数点对齐，便于对个、十、百、千位上的数字进行对比。3 个数字以上，使用千分符作为分隔，便于用户识别，如图 35 所示。




 红包	转入	+10,000.00	 信用卡还款	共还款	544,432,600.00
	转出	-300.00			
 打赏	转入	+10,000.00	 转到银行卡	共转出	77,896,950.00
	转出	0.00			
			 花呗消费	共消费	28,790.00

图 35 数字 / 小数点对齐

4. 图标

(1) 图标网格

使用图标网格可确保图标设计的一致性，从而建立一套明确的图形元素定位规则，如图 36 所示。

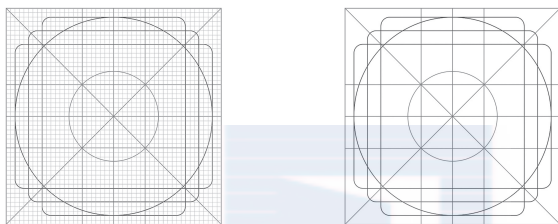
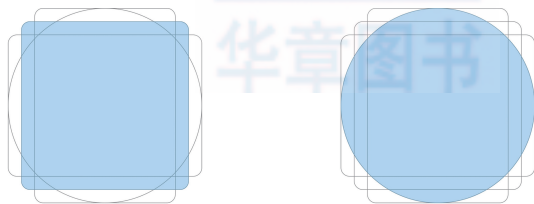


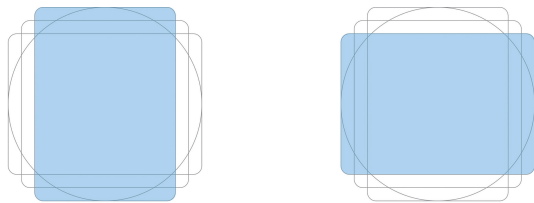
图 36 图标网格

(2) 关键线的形状

关键线的形状是网格的基础。利用这些核心形状作为向导，即可使整个相关产品的图标保持一致的视觉比例，如图 37 所示。



a) 方形、圆形



b) 竖直矩形、水平矩形

图 37 关键线的形状

(3) 产品图标

产品图标是传达品牌内涵的一个重要载体，是一个品牌下产品、服务和工具的一种视觉表现，通过对每个功能属性的调用，确定用 4 种色彩区分各功能，能够以简洁的外形、鲜明的颜色友好地传递产品的核心理念与内涵，如图 38 所示。



图 38 产品图标

5. 适配

(1) iOS 适配

iOS 机型总览如图 39 所示。



图 39 iOS 机型总览

(2) Android 适配

Android 主流机型概览如图 40 所示。

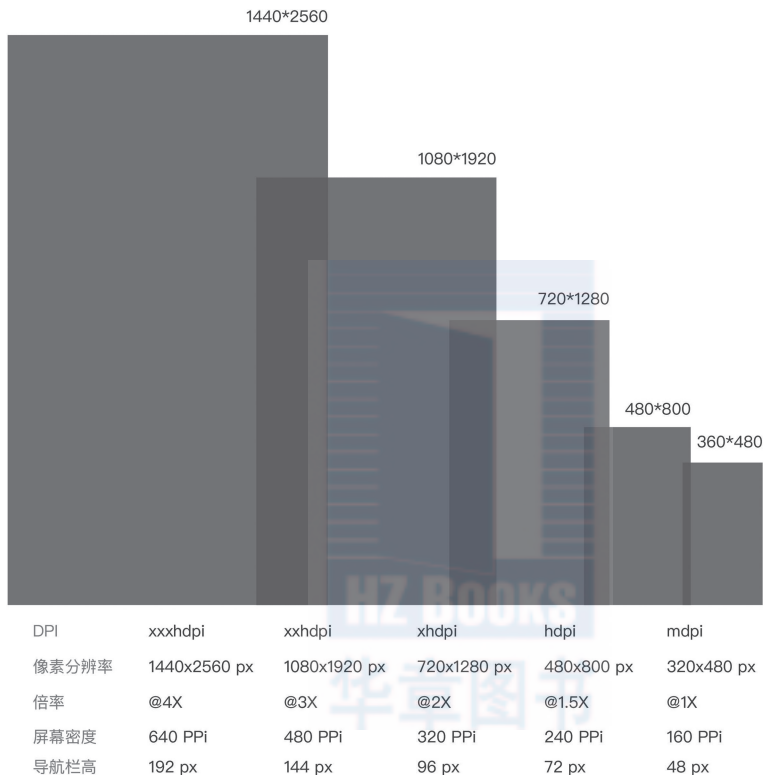


图 40 Android 主流机型概览

App 设计开发必须考虑适配各种屏幕。如何做到交付一套设计稿就可解决适配大、中、小三屏的问题？设计和开发之间采用什么协作模式？一个基本思路如下：

- 选择一种尺寸作为设计和开发基准。
- 定义一套适配规则，自动适配剩下尺寸。

- 特殊适配效果给出设计效果。

第一步：视觉设计阶段，设计师按宽度 750px（iPhone 6）做设计稿，除图片外所有设计元素用矢量路径来做。设计定稿后在 750px 的设计稿上做标注，输出标注图。同时等比放大 1.5 倍生成宽度 1125px 的设计稿，在 1125px 的稿子里切图。

第二步：输出两个交付物给开发工程师：一个是程序用到的 @3X 切图资源，另一个是宽度 750px 的设计标注图。

第三步：开发工程师拿到 750px 标注图和 @3X 切图资源，完成 iPhone 6（375pt）的界面开发。此阶段不能用固定宽度的方式开发界面，得用自动布局（auto layout），方便后续适配到其他尺寸。

第四步：适配调试阶段，基于 iPhone 6 的界面效果，分别向上、向下调试 iPhone 6 plus（414pt）和 iPhone 5S 及以下（320pt）的界面效果。由此完成大、中、小三屏适配。

iOS 设计适配流程如图 41 所示。

第一步：视觉设计阶段，设计师按宽度 1080px 做设计稿，除图片外所有设计元素用矢量路径来做。设计定稿后在 1080px 的设计稿上做标注，输出标注图。同时等比放大 4/3 倍生成宽度 1440px 的设计稿，在 1440px 的稿子里切图。

第二步：输出两个交付物给开发工程师：一个是程序用到的 @4X

切图资源，另一个是宽度 1080px 的设计标注图。

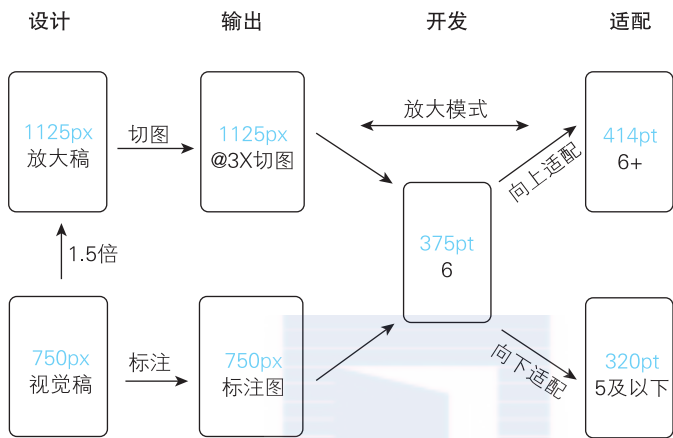


图 41 iOS 设计适配流程

第三步：开发工程师拿到 1080px 标注图和 @4X 切图资源，完成 1080px 的界面开发。此阶段不能用固定宽度的方式开发界面，得用自动布局 (auto layout)，方便后续适配到其他宽度尺寸。

第四步：适配调试阶段，基于 1080px 的界面效果，分别向上、向下调试 1440px 和 720px 及以下的界面效果。由此完成大、中、小三屏适配。

Android 设计适配流程如图 42 所示。

只交付一套设计稿，默认用什么规则来适配？

前文提到适配策略是先选择一种尺寸作为基准设计尺寸，然后通过一套适配规则自动适配到其他尺寸。这套适配规则总结起来就是：文字

流式，控件弹性，图片等比缩放，适配系数。

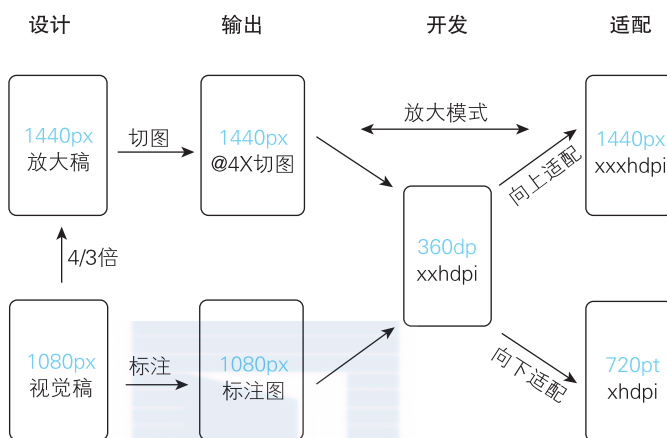


图 42 Android 设计适配流程

- **文字流式**：在不同的设备上，文字大小不变，文字显示的区域产生变化。通常 iOS5 的文字显示区域更长，一行可以展示更多的文字，如图 43 所示。



图 43 文字流式

- **控件弹性**：navigation、cell、bar 等适配过程中垂直方向上高度不变、水平方向宽度变化时，通过调整元素间距或元素右对齐的方式实现自适应。这样屏幕越大，在垂直方向上可以显示越多的内容，发挥大屏幕的优势，如图 44 所示。



图 44 控件弹性

- **图片等比缩放**：当涉及插图、banner 等位图时，这些图片在不同设备中根据屏幕大小等比缩放，保证不变形，如图 45 所示。

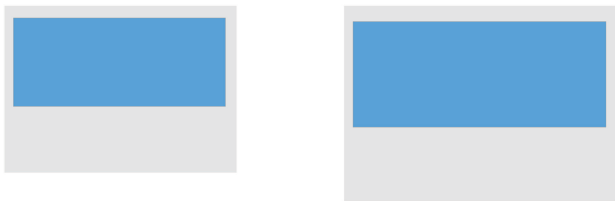


图 45 图片等比缩放

- **适配系数**：同屏幕密度的适配，如 6、5、4s，用到前面 3 个规则就可以了；跨屏幕密度的适配，如 6+、6、3gs，需要用 一个适配系数进行换算以后再用前面 3 个规则，如图 46 所示。

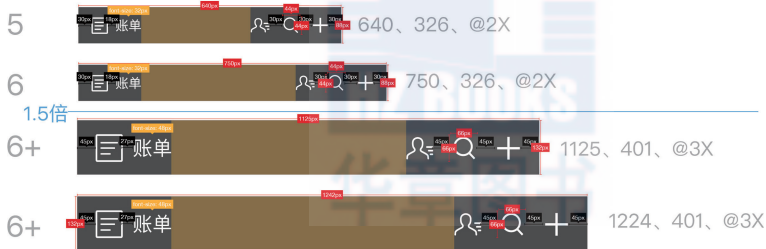


图 46 适配系数实例

- **特殊适配**：有些界面通过以上规则进行适配以后，可能并不能达到设计师理想的效果。这时候需要对这个页面做各个屏幕的设计稿，进行特殊适配。为了更好地展示效果，工作量无疑会成倍地增长。

以支付宝首页为例，90×90dp(1080px 设备上像素为 270×

270px) 的应用宫格，在 1080px 设备上显示一行 4 个，刚好充满屏幕宽。同样的宫格，放在 480px 设备上(480px 设备上像素为 135×135px)，4 个应用占据屏幕宽度 135×4=540px 已经超过屏幕尺寸，放在 1440px 屏幕上(1440px 屏上像素为 315×315px)，4 个应用占宽 315×4=1260px，比 1440px 的屏幕宽度还小一些。

这种情况下就要单独考虑屏幕适配方案了，例如：480px 屏幕缩小应用尺寸，改为 80×80dp；或者单独提供适合 480px 的切片；等等，如图 47 所示。



图 47 特殊适配

第 4 节 控件 & 组件

一致性的原则要求，系统中同样功能的元素、控件、界面应该在样式、交互行为上都保持一致。前面我们建设了设计原则和基础规范，从思维方式和基本元素的层面为一致性打下了基础。如果只有这两部分，对于产品整体的标准化来说还不完整。因为在同样的设计原则和基础规范下，设计出来的导航栏可能不一样。所以，我们还要建立控件以及组件的标准，整个规范才能完整地发挥作用。

那么问题来了，到底页面中哪些控件应该是标准化的？哪些组件应该是统一调用的？哪些东西是固定不变的？哪些东西是可以变化的？如果什么都固定死了，设计师还有发挥的空间吗？

所以，定义好控件、组件的范围，才能把握好标准化和创新之间的微妙平衡。

哪些元素可以定义为控件，哪些页面可以统一成组件？如何界定标准化和设计创新的边界？我们对支付宝内各种类型页面进行了拆分，以是否需要一致作为标准进行归类 and 整理。图 48 所示为典型的页面拆分案例。

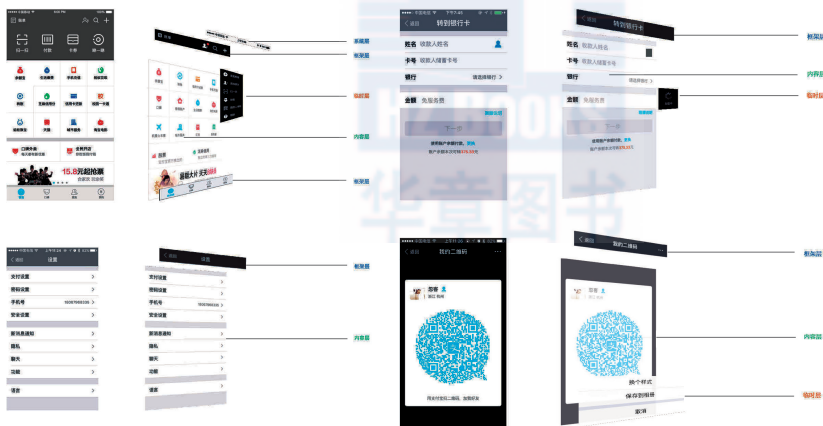


图 48 典型的页面拆分案例

通过对支付宝数百个页面的拆解和分析，我们根据页面元素的稳定性，将所有的页面元素归纳为不同的 5 个层级，并且分别定义每个层级的设计自由度。如图 49 所示。



图 49 层级定义

1. 系统层

系统层——完全调用系统原生的控件和组件，不做任何额外的定义和设计。所有产品和页面都应该是一致的，最大限度符合系统特性，随着系统更新变化而变化。系统层包括状态栏、系统通知、控制控件、系统键盘、手势，如图 50 所示。

2. 框架层

框架层——用于组织页面信息，并且起到导航作用的控件。这部分控件在符合系统体验原则的基础上，应该反映支付宝的品牌特点。所以框架层需要根据支付宝自身的特点定制和开发，但是它在支付宝内部应该是一致的。这部分控件包括导航栏、Tab 栏、分段控件、工具栏，如图 51 所示。

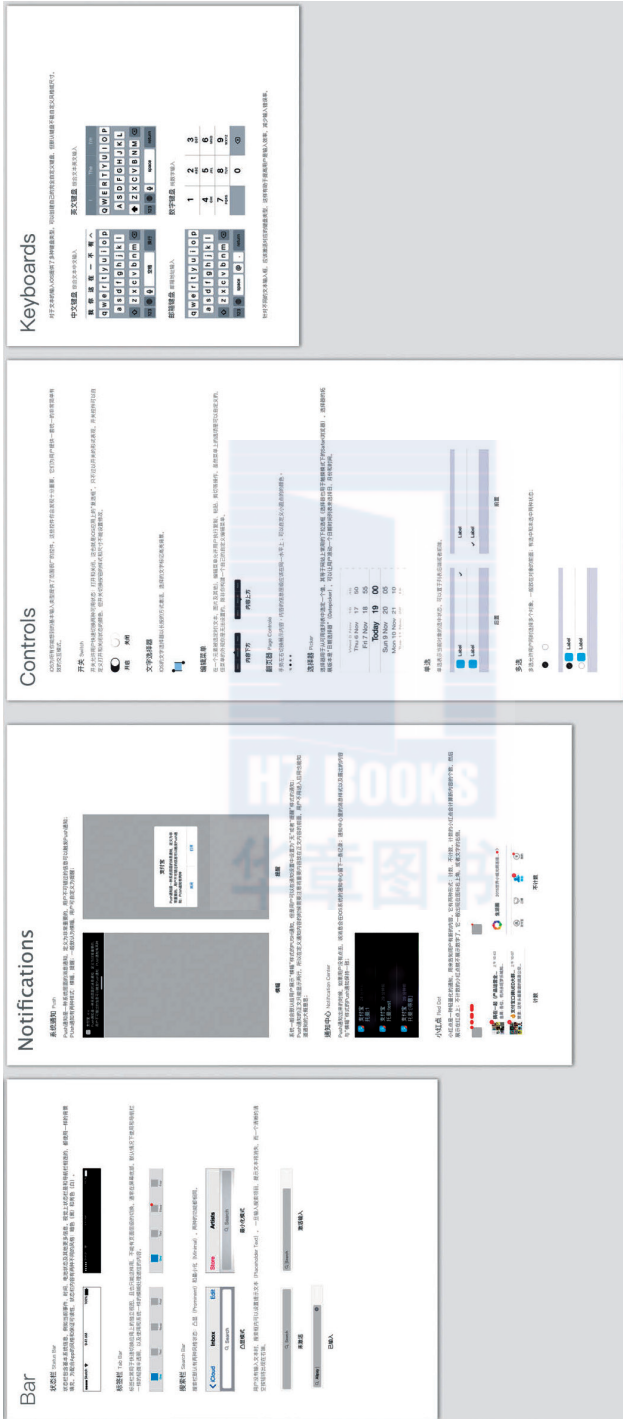


图 50 系统层定义示例

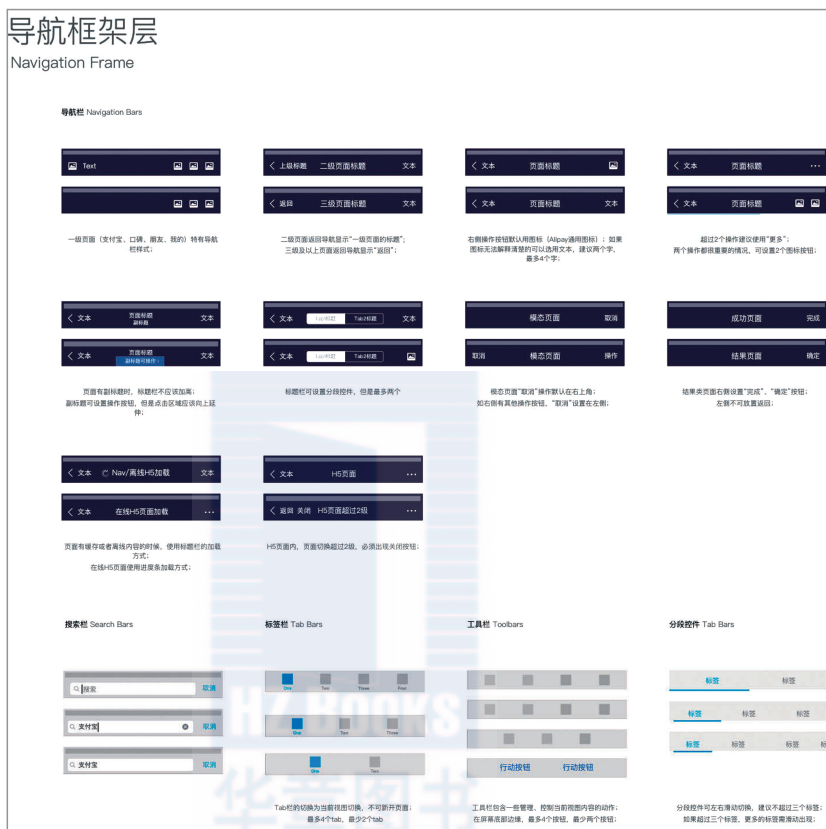


图 51 框架层定义示例

3. 临时层

临时层——页面中临时出现的浮层和内容, 在调整符合系统交互特性的基础上, 样式根据支付宝特点定制和开发。支付宝内部需要保持体验的一致性。临时层包括活动视图、活动菜单、POP 菜单、弹框、Toast、选择器、临时公告等。如图 52 所示。

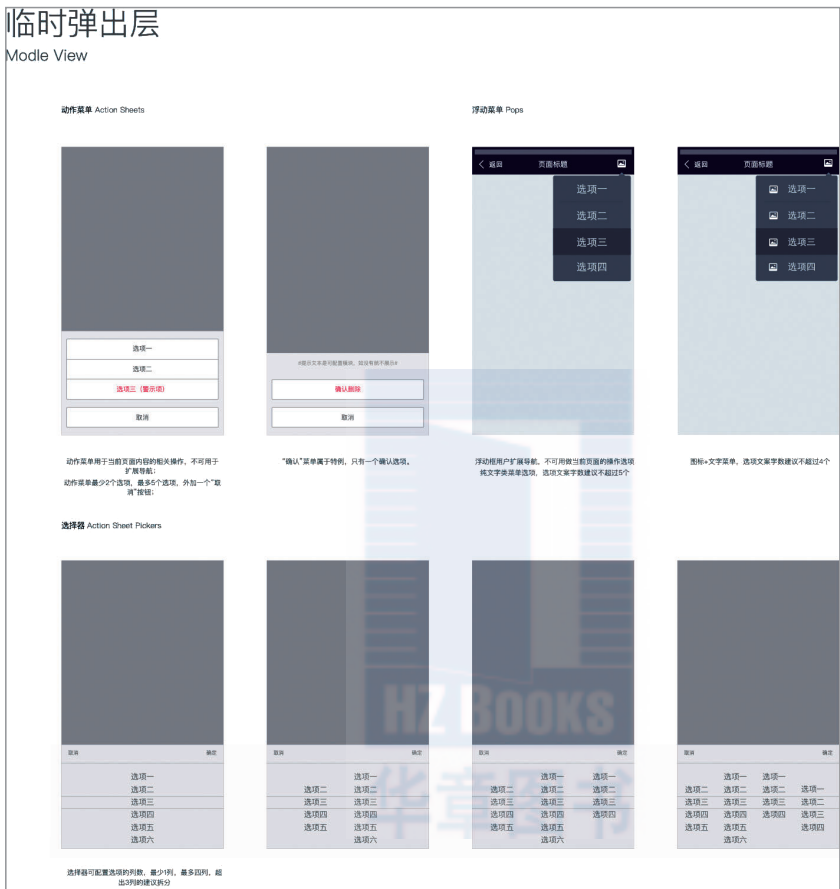


图 52 临时层定义示例

4. 内容层

内容层——页面中剩下的内容，跟页面内容的特点相关，每个页面可能都不同，是设计师发挥空间最大的部分。但是一些具有共性的控件，如列表项、按钮、输入框等可以抽离出来，做成标准化，如图 53 所示。



图 53 内容层定义示例

第 5 节 输出规范

世上本没有路，走的人多了，也便成了路。——鲁迅

规范如果没有人使用，那就是一堆文档。当然，为了保障产品整体的体验，我们可以采用行政手段，强制要求团队成员阅读并且遵守规范

文档。这全靠每个人的记忆力和自觉性，并不能很好地达到我们想要的效果。因为每个人对同一个东西的理解和执行度都是不一样的。

强制的方法不好，我们就得换一个角度，从每个人的利益出发，让大家都乐意使用规范。这个利益点就是“效率”，我们要让规范帮助大家提升工作效率、沟通效率。

1. 规范文档 & UIKit

怎样的规范能提升设计师的工作效率，而不让他们觉得这只是一种约束呢？

考虑到设计师的工作特点，我们将设计规范分为规范文档和 UIKit 两部分输出。规范文档是最全的文档，里面详细阐述了设计原则、基础规范，以及每个控件的样式和使用规则。这部分文档需要设计师阅读并且理解里面的规则。UIKit 则是控件样式的集合，我们提供和维护最新的样式源文件。设计师在工作中只需要复制和拖动，不必重复绘制和单独设计，节省了大家的时间。

2. 开发基础控件库

设计规范仅仅在设计师团队中推广是不够的，因为设计最后的实施是在开发环节。我们以效率和成本作为突破口，在开发团队中推广和执行我们的标准化。开发人员每天加班写代码，所以他们是最喜欢标准化的了。所以，建立开发的基础控件库，让各个产品的开发都可以直接调用和配置参数，这是规范和标准化执行的终极环节。

总结

规范是死的，设计是活的。设计文档和标准控件是死的，设计思想是活的。标准化建设的过程中一定要把握好这个平衡，不能让规范制约了创新。建立团队统一的设计价值观，界定一个好的设计框架，让设计师在一致的范围里创造是设计规范的价值所在。所以，设计原则、规范理论基础等的提炼和建设在规范体系里应该得到我们的重视。“戴着镣铐跳舞”讲的正是这个道理。

