IOS

1. iOS设计主题
   1. 三个主要的主题区别于其他平台：
      1. Clarity——清晰。
         1. 清晰的文本、图标
         2. 点缀精巧且恰当
      2. Deference——尊重
         1. 流畅的界面和清晰没关的界面可帮助人们理解内容并与之互动。
         2. 少使用边框，渐变和阴影，可以使得界面轻盈。
      3. Depth——深度
2. 设计原则
   1. 诚信审美原则
      1. 一个帮助人们执行严肃任务的app，可以通过不显眼的图形，标准空间和可预测行为保持人们的专注。
      2. 另一方面，沉浸式的app，如游戏，可提供迷人的外观鼓励探索。
   2. 一致性原则
      1. 标准的界面元素、图标、文本样式、统一的术语。
   3. 直接控制原则
      1. 操控界面得到行为可见的结果。
   4. 反馈原则
      1. 内置的iOS应用可响应每个用户的操作，进度指示器，动画，声音等等。
   5. 隐喻原则
      1. 与现实世界的相似性，如切换开关，翻阅书籍，会让人们更快地学习。
   6. 用户控制原则
3. 接口基础？
   1. Bars
   2. Views
   3. Controls

App Architecture

1. 无障碍篇
   1. 为图像，图标和界面元素提供替代文本标签。
   2. 响应可访问性首选项。粗体和大文本
   3. 开启辅助功能。对比度，反转颜色，降低透明度。
   4. 隐藏式字幕和音频说明。
   5. 使用足够的颜色对比度。
      1. 图标和文本可能与BG混合在一起，在线颜色对比度计算器，7：1是首选。
2. Loading：当内容加载时，空白或静态屏幕可能会使您的应用程序被冻结，导致他人离开。
   1. 让加载更侵袭。
      1. 至少显示微调器。更好的方式是显示明确进度，以便用户可以判断他们将等待多长时间。
   2. 尽快展示内容，越快越好。
      1. Show the screen immediately, and use placeholder text, graphics, or animations to identify where content isn't available yet. Replace these placeholder elements as the content loads. Whenever possible, preload upcoming content in the background, such as while an animation is playing or the user is navigating a level or menu.
   3. Educate or entertain people to mask loading time.
      1. 考虑显示有关游戏玩法，娱乐视频序列或有趣的占位符图形的提示。
   4. Customize loading screens.
3. Modality
   1. Minimize the use of modality.通常，人们更喜欢非线性方式互动。Modality只有在获得注意，或者必须完成或放弃任务，保存重要数据时才去使用。
   2. Provide an obvious and safe way to exit a modal task.
   3. Keep modal tasks simple, short, and narrowly focused.不要让Modality太复杂，否则可能会让人们暂停任务。
   4. Display a title that identifies a task, if necessary.
   5. Reserve alerts for delivering essential—and ideally actionable—information.
   6. Respect notification preferences.
   7. Don’t display a modal view above a popover.
   8. Coordinate modal view appearance with your app.
   9. Choose a modal view style that works in the current environment.
   10. Choose a modal transition style that makes sense in your app.
4. Navigation
   1. Hierarchical Navigation——分层导航
      1. 每个屏幕做一个选择，直到到达目的地。要前往另一个目的地，您必须回溯您的步骤或从头开始并做出不同的选择。设置和邮件使用此导航样式。
   2. Flat Navigation——平面导航（apple music & App Store）
   3. Content-Driven or Experience-Driven Navigation——内容驱动或体验驱动的导航
   4. 导航原则
      1. Alaways provide a clear path
      2. Design an information structure that makes it fast and easy to get to content.
         1. 以需要最少数量的点击，滑动和屏幕的方式组织您的信息结构。
      3. Use touch gestures to create fluidity.
         1. 用最小摩擦力轻松移动页面，例如，人们可以从屏幕侧面滑动以返回上一个屏幕。
      4. Use standard navigation components.
         1. 尽可能使用标准导航控件，例如页面控件，选项卡栏，分段控件，表视图，集合视图和拆分视图。
      5. Use a navigation bar to traverse a hierarchy of data.
         1. 导航栏的标题可以显示层次结构中的当前位置，后退按钮可以轻松返回到先前的位置。
      6. Use a tab bar to present peer categories of content or functionality.
         1. 标签栏可让人们快速轻松地在不同类别之间切换，无论当前位置如何。
      7. Use a page control when you have multiple pages of the same type of content.
         1. 页面控件清楚地传达可用页面数和当前活动页面数。Weather应用程序使用页面控件来显示特定于位置的天气页面。
5. Onboarding
   1. 提供启动屏幕
   2. 选择好合适的方向
   3. Get to the action quickly.
      1. 避免显示启动画面，菜单和说明。让人们直接潜入，如果app需要教程，需要提供跳过方式
   4. Avoid asking for setup information up front.——避免实现询问设置信息
   5. Restore the previous state when your app restarts.
      1. 不要让人们回溯到您应用中的先前位置。保留并恢复您应用的状态，以便他们可以从中断的地方继续。
   6. Don’t ask people to rate your app too quickly or too often.
   7. Don’t encourage rebooting.
6. Requesting Permission
   1. Request personal data only when your app clearly needs it.
      1. 确保仅在人们使用明确需要个人数据的功能时才会发出权限请求。例如，应用可能仅在激活位置跟踪功能时请求访问当前位置。
   2. Explain why your app needs the information.
      1. 该应用程序会在夜间记录您以检测打鼾声音。 YES
      2. 打开麦克风访问。 XXX
   3. Request permission at launch only when necessary for your app to function.
   4. Don’t request location information unnecessarily.
   5. Use the system-provided alert.
7. 设置
   1. Infer what you can from the system.
      1. 如果您需要有关用户，设备或环境的信息，请尽可能向系统查询，而不是询问用户。例如，不要让某人输入他们的邮政编码，以便您可以提供本地选项，而是要求获得使用其当前位置的权限。如果用户拒绝访问其信息，则优雅地回退到手动输入。
   2. Thoughtfully prioritize configuration options within your app.
      1. 您的应用程序的主屏幕是一个选择必要或经常更改的好地方。辅助屏幕更适合仅偶尔更改的选项。
   3. Expose infrequently changed configuration options in Settings.
      1. “设置”应用是在整个系统中进行配置更改的中心位置，但人们必须离开您的应用才能到达目的地。在您的应用程序中直接调整设置会更方便。如果必须提供很少需要更改的设置，请参阅“ 首选项和设置编程指南”中的“ 实现iOS设置包”以获取开发人员指南。
   4. Provide shortcuts to Settings when appropriate.
      1. 如果您的应用包含将用户定向到“设置”的文本，例如“转到设置> MyApp>隐私>位置服务”，则会提供一个自动打开该位置的按钮。

用户交互

1. 3D Touch
   1. Use peeking to provide live, content-rich previews.
   2. Design big-enough peek views.
   3. Adopt Peek and Pop consistently.
      1. 如果您在某些地方支持Peek和Pop而不支持其他地方，人们将不知道他们可以在哪里使用该功能，并且可能认为您的应用或其设备存在问题。
   4. Avoid displaying button-like elements in a peek view.
   5. Allow every peek to be popped.
   6. Don’t enable peeking and an edit menu for the same item.
   7. Provide action buttons when appropriate.
   8. Avoid providing an action button that opens a peeked item.
   9. Don’t make peek the only way to perform item actions.
2. 数据输入
   1. When possible, present choices.
      1. 使数据输入尽可能高效。例如，考虑使用选择器或表而不是文本字段，因为从预定义选项列表中选择比键入响应更容易。
   2. Get information from the system whenever possible.
      1. 不要强迫人们提供可以自动或经用户许可收集的信息，例如联系人或日历信息。
   3. Providing good defaults minimizes decision making and speeds up the process.
      1. 在可能的范围内，预填充具有最可能值的字段。提供良好的默认值可以最大限度地减少决策制定并加快流程。
   4. Enable advancement only after collecting required values.
      1. 在启用“下一个”或“继续”按钮之前，请确保所有必填字段都有值。使用按钮的启用作为视觉提示，是时候继续。
   5. Dynamically validate field values.
      1. 填写冗长的表格后，如果必须返回并纠正错误，那将是令人沮丧的。只要有可能，请在输入后立即检查字段值，以便用户立即纠正它们。
   6. Require field values only when necessary.
      1. 仅将必填字段用于真正需要继续操作的信息。
   7. Ease navigation through value lists.
      1. 考虑按字母顺序或以另一种逻辑方式对值列表进行排序，以便于快速扫描和选择。
   8. Show a hint in a text field to help communicate purpose.
      1. 文本字段可以包含占位符文本 - 例如“电子邮件”或“密码” - 当字段中没有其他文本时。当占位符文本足够时，请勿使用单独的标签来描述文本字段。

系统功能

1. 通知
   1. 通知形式
      1. Banner. 当设备正在使用时，在屏幕顶部出现几秒钟，然后消失。
      2. Alert. 在设备使用时出现在屏幕顶部并保持在那里直到手动关闭。
   2. 提供有用的信息性通知。
      1. 使用完整的句子，句子大小写，正确的标点符号，并且不要截断您的消息 - 系统会在必要时自动执行此操作。
   3. 即使用户没有响应，也不要为同一件事发送多个通知。
   4. 请勿包含您的应用名称或图标。系统会在每个通知的顶部自动显示此信息。
   5. 提供描述性文本，以便在隐藏通知预览时显示。
   6. 提供声音以补充您的通知。
      1. 当他们不看屏幕时，声音是引起别人注意的好方法。待办事项列表应用程序可能会播放警报声，例如，在执行重要任务时。您的应用可以使用自定义声音或内置警报声音。如果您使用自定义声音，请确保它简短，独特且专业。
   7. 考虑提供详细视图。
      1. 通知详细信息视图提供有关通知的详细信息. 此视图应包含有用的信息，可识别，并且感觉就像是应用程序的自然扩展。它可以包含照片，视频和其他内容，并可以在显示时动态更新。例如，共乘应用程序可以在此处显示一张地图，显示接近当前位置的汽车。
   8. 提供直观，有益的行动。
   9. Badging（徽标）
      1. 使用标记来补充通知，而不是表示关键信息
      2. 使用标记仅用于通知目的。
      3. 保持徽标最新。
2. 评分和评论
   1. 仅在用户证明与您的应用互动后才要求评分。
   2. 不要打断用户，特别是当他们执行时间敏感或压力很大的任务时。
   3. Don’t be a pest.

Visual Design

1. 适应性和布局
   1. 屏幕大小指南：<https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/>
   2. Auto Layout
   3. Layout Guides and Safe Area
      1. 布局指南定义了实际上不会在屏幕上显示的矩形区域，但有助于内容的定位，对齐和间距。该系统包括预定义的布局指南，可以轻松地在内容周围应用标准边距并限制文本宽度以获得最佳可读性。您还可以定义自定义布局指南。
   4. Size Classes
      1. 大小类是根据大小自动分配给内容区域的特征。系统定义了两个大小类，regular（表示扩展空间）和compact（表示约束空间），它们描述视图的高度和宽度。
   5. General Layout Considerations
      1. Ensure that primary content is clear at its default size.
         1. 确保主要内容以其默认大小能看清。除非他们选择更改大小，否则人们不应该水平滚动来阅读重要文本，或者缩放以查看主要图像。
      2. 在整个应用中保持整体一致的外观。
      3. Use visual weight and balance to convey importance.
         1. 使用视觉重量和平衡来传达重要性。大件物品吸引眼球，比看起来比较小的物品更重要。较大的物品也更容易点击，这在应用程序用于分散注意力的环境时尤为重要。通常，将主要项目放置在屏幕的上半部分中，并且在从左到右的读取上下文中 - 靠近屏幕的左侧。
      4. 使用对齐可以简化扫描并与组织和层次结构进行通信。
         1. 对齐使应用程序看起来整洁有序，帮助人们在滚动时集中注意力，并使查找信息变得更加容易。缩进和对齐还可以指示内容组如何相关。
      5. If possible, support both portrait and landscape orientations.
         1. 人们更喜欢在不同的方向使用应用程序，因此最好能够满足这种期望。
      6. Be prepared for text-size changes.
         1. 人们期望大多数应用在设置中选择不同的文字大小时进行响应。
      7. 为交互式元素提供充足的触摸目标。
         1. 尝试为所有控件保持44pt x 44pt的最小可点击区域。
      8. Preview your app on multiple devices.
      9. 在较大的设备上显示文本时应用可读性边距。
   6. Adapting to Changes in Context
      1. 在上下文变化期间保持对当前内容的关注。
         1. 内容是您的最高优先级。在环境变化时改变焦点可能令人迷茫和令人沮丧，并且可能让人觉得他们失去了对应用程序的控制。
      2. 避免无偿的布局更改。
         1. 当有人旋转设备时，整个布局不必更改。例如，如果您的应用以纵向模式显示图像网格，则不必在横向模式下显示与列表相同的图像。相反，它可能只是调整网格的尺寸。
      3. 确保您的应用在iPad上运行，而不仅仅在iPhone上运行。
      4. 重用现有艺术作品时请注意纵横比差异。
         1. 不同的屏幕尺寸可能具有不同的宽高比，导致图稿出现裁剪，信箱或邮筒。确保所有显示尺寸都能保留重要的可视内容。
   7. 设计全屏体验
      1. 扩展可视元素以填充屏幕。
      2. 避免将交互式控件明确放置在屏幕的最底部和角落中。
         1. 人们使用显示屏底部的滑动手势来访问主屏幕和应用切换器等功能，这些手势可能会取消您在此区域中实现的自定义手势。屏幕的远角可能是人们难以到达。
      3. 插入必要内容以防止剪裁。
         1. 通常，内容应该居中且对称插入，因此它在任何方向上看起来都很好，没有被圆角修剪，不被传感器外壳隐藏，并且不会被用于访问主屏幕的指示器遮挡。
      4. Inset full-width buttons.
         1. 延伸到屏幕边缘的按钮可能看起来不像按钮。尊重全宽按钮两侧的标准UIKit边距。当屏幕底部出现圆角并与安全区域的底部对齐时，屏幕底部出现的全宽按钮效果最佳 - 这也确保它不会与Home指示灯冲突。
      5. 请勿掩盖或特别注意按键显示功能。
         1. 不要试图通过在屏幕的顶部和底部放置黑条来隐藏设备的圆角，传感器外壳或用于访问主屏幕的指示器。
      6. 请注意状态栏的高度。
         1. 全屏iPhone上的状态栏比旧款iPhone更高。如果您的应用假设状态栏高度为状态栏下方的定位内容，则必须更新您的应用以根据用户的设备动态定位内容。请注意，当语音录制和位置跟踪等后台任务处于活动状态时，全屏iPhone上的状态栏不会更改高度。
      7. 如果您的应用当前隐藏了状态栏，请重新考虑全屏iPhone的决定。
         1. 与旧款iPhone相比，全屏iPhone的内容垂直空间更大，状态栏占据了您的应用可能无法充分利用的屏幕区域。状态栏还显示人们认为有用的信息。它只应隐藏以换取附加值。
2. Animation
   1. 整个iOS中美丽，微妙的动画构建了人与屏幕内容之间的视觉连接感。如果使用得当，动画可以传达状态，提供反馈，增强直接操控感，并帮助用户可视化他们的行为结果。
   2. 明智地使用动画和动作效果。
      1. 不要为了使用动画而使用动画。过度或无偿的动画可能会让人感到分离或分心，特别是在没有提供身临其境体验的应用中。iOS使用运动效果（例如视差效果）在主屏幕和其他区域创建深度感。这些效果可以增加理解和享受，但过度使用它们会使应用程序感到迷失方向并且难以控制。如果实现运动效果，请始终测试结果以确保它们正常工作。
   3. 力求现实主义和信誉。
      1. 人们倾向于接受艺术许可，但是当运动没有意义或似乎违反物理定律时，他们会感到迷失方向。
   4. 使用一致的动画。
      1. 熟悉，流畅的体验让用户保持参与。他们习惯于iOS中使用的微妙动画，例如平滑过渡，设备方向的流畅变化和基于物理的滚动。除非您正在创建沉浸式体验，例如游戏，否则自定义动画应与内置动画相媲美。
   5. 使动画可选。
      1. 在辅助功能首选项中启用减少运动的选项时，您的应用应最小化或消除应用程序动画。
3. Branding
   1. 成功的品牌塑造不仅仅是为您的应用添加品牌资产。优秀的应用程序通过智能字体，颜色和图像决策表达独特的品牌标识。提供足够的品牌以在您的应用中为人们提供上下文，但不要太多，以至于它会分散注意力。
   2. 融合精致，不引人注目的品牌。
   3. 不要让品牌妨碍优秀的应用程序设计。
   4. 推迟内容而不是品牌推广。
      1. 在屏幕顶部显示一个持久栏，只显示品牌资产，这意味着查看内容的空间较小。相反，请考虑使用较少侵入性的方式来实现品牌，例如使用自定义颜色方案或字体，或巧妙地自定义背景。
   5. 抵制在整个应用中显示徽标的诱惑。
      1. 避免在整个应用中显示徽标，除非有必要提供上下文。这在导航栏中尤其重要，其中标题更有用
   6. App Store营销指南：<https://developer.apple.com/app-store/marketing/guidelines/>
4. [颜色](https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/color/)
   1. 颜色是一种传递活力，提供视觉连续性，传达状态信息，响应用户操作提供反馈以及帮助人们可视化数据的好方法。在选择浅色和深色背景下单独和组合的应用色调颜色时，请查看系统的配色方案以获得指导。
   2. 明智地使用颜色进行交流。
      1. 当谨慎使用时，提高注意重要信息的颜色力量会增强。例如，当出于非关键原因在应用程序中的其他地方使用红色时，警告人们关键问题的红色三角形变得不那么有效。
   3. 在整个应用中使用补色。
      1. 应用中的颜色应该很好地协同工作，而不是冲突或分散注意力。例如，如果粉彩对于应用程序的样式至关重要，请使用一组协调的粉彩。
   4. 通常，选择与您的应用徽标协调的有限调色板。
      1. 微妙地使用颜色是传达品牌的好方法。
   5. 考虑选择一种关键颜色，以指示整个应用程序的交互性。
      1. 在Notes中，交互元素为黄色。在日历中，交互式元素为红色。如果您定义表示交互性的键颜色，请确保其他颜色不与其竞争。
   6. 避免对交互式和非交互式元素使用相同的颜色。
      1. 如果交互式和非交互式元素具有相同的颜色，则人们很难知道在哪里挖掘。
   7. 使用足够的颜色对比度。
      1. 您的应用中对比度不足会让每个人都难以阅读内容。例如，图标和文本可能与背景混合在一起。在线颜色对比度计算器可帮助您准确分析应用中的颜色对比度，以确保其符合最佳标准。力争最小对比度为4.5：1，尽管7：1是首选，因为它符合更严格的可访问性标准。
5. 文案-术语- Terminology
   1. 您应用中的每个字都是与用户对话的一部分。使用此对话可以帮助他们在您的应用中感到舒适。
   2. 使用熟悉，易懂的单词和短语。
      1. 技术可能令人生畏。避免人们可能不理解的首字母缩略词和技术术语。使用您对受众的了解来确定某些字词或短语是否合适。一般而言，吸引所有人的应用应该避开高技术语言。这种语言可能适用于针对更高级或技术人群的应用。
   3. 保持界面文本清晰简洁。
      1. 人们可以快速轻松地吸收简短直接的文本，并且不喜欢被迫阅读长篇文章来完成任务。确定最重要的信息，简洁地表达，并突出显示，以便人们不必过多地阅读以找到他们正在寻找的内容或找出下一步该做什么。
   4. 适当地识别互动元素。
      1. 人们应该能够一眼就看出元素的作用。标记按钮和其他交互元素时，请使用动作谓词，例如“连接”，“发送”和“添加”。
   5. 避免使用可能听起来光顾的语言。
      1. 避免我们，我们，我和我（例如“我们的教程”和“我的锻炼”）。他们有时被解释为侮辱或光顾。
   6. 力求非正式，友好的语气。
      1. 一种非正式，平易近人的风格与您在午餐时与人交谈的方式相呼应。偶尔使用收缩，您和您直接对用户说话。
6. 印刷
   1. San Francisco (SF) is the system typeface in iOS. The fonts of this typeface are optimized to give your text unmatched legibility, clarity, and consistency.
   2. 强调重要信息。
      1. 使用字体粗细，大小和颜色突出显示应用中最重要的信息。
   3. 如果可能，请使用单一字体。
      1. 混合几种不同的字体可能会使您的应用程序看起来支离破碎和草率。考虑使用一种字体和几种字体变体和大小。
   4. 尽可能使用内置文本样式。
      1. 内置的文本样式允许您以视觉上不同的方式表达内容，同时保持最佳的易读性。这些样式基于系统字体，允许您利用关键的排版功能，例如动态类型，它可以自动调整跟踪并引导每种字体大小。