



# 湖北经济学院 毕业论文（设计）

题 目（中文）：基于渐进式网页应用技术的植物图鉴应用开发

（英文）：Progressive Web Apps development-Plant Illustration APP

专 业：信息管理与信息系统（中美合作）

院 系：信息管理与统计学院

年 级：2015 级

学 号：15150255

姓 名：朱懿帛

指导教师：彭瑞卿

职 称：副教授

湖北经济学院教务处 制

## 目录

|                       |    |
|-----------------------|----|
| 一. 研究的背景及意义 .....     | 5  |
| (一) 研究背景 .....        | 5  |
| (二) 研究问题的提出 .....     | 6  |
| (三) 研究目的 .....        | 6  |
| (四) 研究的意义 .....       | 7  |
| 二. 国内外研究状况综述 .....    | 8  |
| (一) 国外研究状况 .....      | 8  |
| (二) 国内研究状况 .....      | 11 |
| 三. 植物图鉴移动应用程序开发 ..... | 14 |
| (一) 应用程序开发过程 .....    | 14 |
| (二) 系统的总体架构: .....    | 15 |
| (三) 渐进功能的实现: .....    | 16 |
| 四. 总结与展望 .....        | 23 |
| 致谢 .....              | 25 |
| 参考文献 .....            | 26 |

## 摘要

随着移动应用开发技术的不断发展，跨平台形式的移动应用开始出现，基于 Android 和 iOS 两大平台的传统开发应用形式开始受到挑战，其中渐进式网页应用技术作为当下新兴的跨平台开发技术开始受到人们的关注。本文先介绍了这项技术出现的背景，并介绍了渐进式网页应用技术的实质，然后分析了国内外对于这项技术研究的差距，最后应用渐进式网页应用技术完成植物图鉴 APP 的开发，用以展现渐进式网页应用技术的本质。

**关键词：**移动应用开发，网页应用程序，渐进式应用程序技术

## **Abstract**

With the continuous development of mobile application technology, cross-platform mobile applications began to emerge, traditional application development forms based on Android and iOS platforms have been challenged. Among them, progressive web app technology began to attract people's attention as the emerging cross-platform technology. This paper first introduces the background of this technology, and introduces the essence of progressive web app technology, then analyzes the gap between this technology research in China and abroad, and finally uses the progressive web app technology to complete the development of plant illustration APP to demonstrate the essence of progressive web app technology.

**Keywords:** mobile application development, Web APP, PWA technology,

## 一. 研究的背景及意义

### （一）研究背景

通过中国互联网络信息中心（CNNIC）发布的第 41 次《中国互联网络发展状况统计报告》<sup>[1]</sup>，截至 2017 年 12 月，中国手机网民有 7.53 亿，选择手机上网的人从 2016 年的 95.1% 已经提升至 97.5%。PC 端的使用率不断下降，手机正在不断挤占其他上网设备的使用。

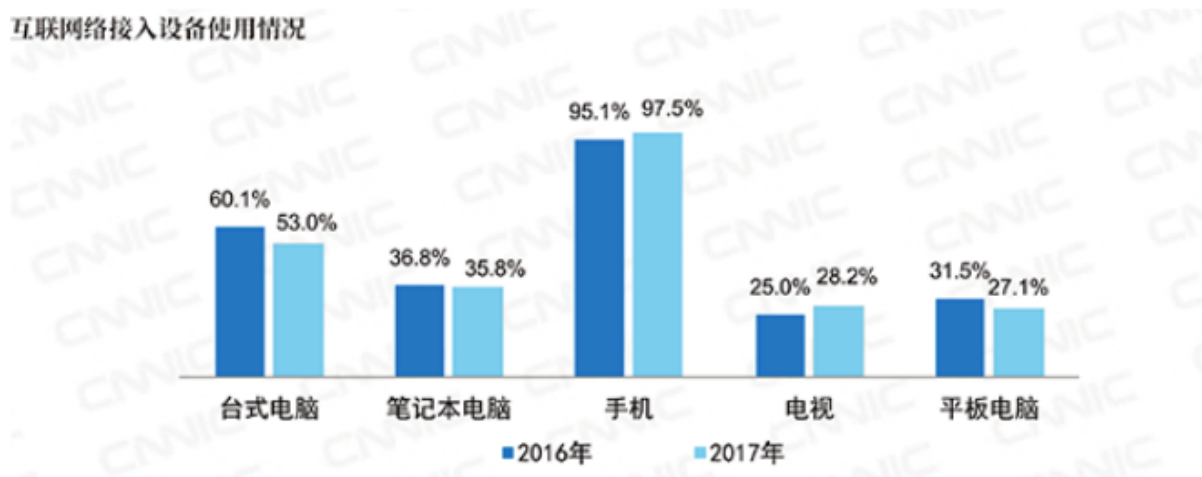


图 1 互联网络接入设备使用情况

由此可见中国的网民在不断的增加，与此同时手机等移动端的上网频率在不断上升并且遥遥领先，从而推动了移动端应用（app）的大幅增长。Taylor<sup>[2]</sup>将移动应用定义为“在移动设备上运行并执行从银行业务到游戏和网络浏览的任务的小程序”。移动程序的激增导致全球智能手机消费者的数量急剧增加。因为移动应用程序易于使用，所以在世界各地都很受欢迎。2017 年全年移动应用下载量接近 2000 亿次，比去年同期相比增长 32%。

在全球范围内，应用程序用户使用应用程序的移动时间比使用 Web 浏览器要多很多。与传统的互联网运营格式相比，移动应用程序在使用中已经超越了基于网站的服务。Gupta<sup>[3]</sup>指出，平均而言，用户使用应用程序花费了 82% 的移动分钟，而使用网络浏览器则只占其移动分钟数的 18%。而移动应用的关键特性是移动性，独特性和速度<sup>[4]</sup>。

但是随着新技术的不断发展，网络浏览器与应用程序之间的界限开始变的模糊，渐进式 web app（Progressive web apps）的开发应运而生，PWA 意图让 Web 在保留其

本质（开放平台、易于访问、可索引）的同时，在离线、交互、通知等方面达到类似 App 的用户体验。

## （二）研究问题的提出

渐进式网页应用开发技术是一项新兴的技术，从开始到现在不到 4 年，而在中国这项技术才刚刚开始起步，本着对网页开发的兴趣和对新技术的不断追求，决定探索这项新技术的实质并通过自主学习做出自己的植物图鉴应用。

## （三）研究目的

现如今移动应用的发展风生水起，各色应用层出不穷，人们也非常习惯使用移动应用来完成日常生活中的娱乐工作和消费行为。但是移动应用也有着许多其本身所带的问题。

- 1) 占内存。移动应用需要全部下载到手机中使用，并且需要持续的升级，这也就说明移动应用将占用极大的内存并且对内存的占有会持续的扩大。
- 2) 下载时间长。这一问题与占内存相辅相成，现如今网络的稳定与高速还不能充分的保证，也就是说在低效的网络环境下，一个大型的 app 将花上几小时甚至一天的时间完成下载，这会严重的影响用户的体验。
- 3) 地域限制严重。由于网络环境在不同的地方是不一样的，大城市可以享受到高效的网络，而小村落的网速却是远远无法保证的，这时如何让大家都可以享受到网络带来的便利就显得尤为重要了。
- 4) 经济方面。现在还处于流量收费的阶段，一个大型的移动应用必然会需要更高的费用，但是根据研究显示，人们往往只用到了移动应用 20% 的功能，也就是说人们为那 80% 付出的费用是无效而浪费的。同时手机的内存也在某些程度上决定了手机的价格，内存越大的手机越贵，但是经济条件不好的人们是无法负荷昂贵的手机的，这时他们对移动应用的使用也就受到了限制。

而渐进式网页应用的开发就可以很好的解决这个问题，它的体积比移动应用小成了百上千倍，也就是一个网页的体积，但却可以完成与移动端应用相同的功能。而我此项研究的目的：

- 1) 将这项强大的技术推广到中国，让更多的人，特别是处于网速不佳地区的人受益。

- 2) 做出在渐进式网页应用这项技术下的植物图鉴 app，描述技术开发原理，并且通过实例展示渐进式网页应用的效果。

#### （四）研究的意义

在渐进式网页应用的植物图鉴 app 开发中，有两点是富有意义的：

(1)渐进式网页开发这项新技术。PWA 作为一项新兴的技术能够显著提高应用加载速度、甚至让 web 应用实现离线环境使用；同时让 web 应用能够像客户端应用一样被添加到主屏、并且全屏执行；并进一步提高了 web 应用与操作系统集成能力。

简而言之，Progressive Web Apps 渐进式的网页应用程序是基于互联网的新型应用（使用 HTML，CSS，和 JavaScript 写出来的 App）。也就是与 App 使用方式，功能相同的网页。随着技术的发展，网页的功能将越来越强大，它不再被局限在笔记本和台式机中，而将被应用于所有的设备。

(2)App 本身而言，这是一个帮助人们培养植物的 app。在生活中我发现大家没事都爱养个花花草草，但却没有一个专门的 App 给出详细的培养意见，比如这种植物喜阳，如何浇水，出现问题时如何解决。我觉得关于植物的问题不是买回来的时候那个小卡片上的几个字就能说明白的，而且大部分人买植物是真的想养活它，并不只为了观赏几天。所以详细而专业的植物图鉴是非常必要的。

这项技术的好处是显而易见的：

从用户的角度来说：

- 1) 跨平台。这个 App 只需网络就能连接不再需要下载在手机中。这意味着所有的信息是被储存在云端的，是否基于 apple 平台或是 Android 平台将变得不再重要。所以无论更换什么设备，信息都是同步更新不会丢失。
- 2) 释放内存。因为无需下载整个 App，而是直接通过 URL 网页访问。这就意味着 App 将不再占用大部分的内存。
- 3) 安全。与传统的网络应用相比，PWA 更安全，因为它们始终通过 HTTPS 提供服务。这可以确保应用程序可以防篡改并且不会被破坏。

从商家角度：节省人力物力财力。相比于开发网页，开发 App 要花将近 10 倍多的时间，如果想要开发在不同平台上的 App 花费就会更高。而这项技术的出现意味着商家有了更多的选择，他们可以选择只开发自己的网页，因为网页也可以拥有与本地 App 相似的功能，同时商家可以节省出一大部分的精力，从而更专注在网页开发

上。

## 二. 国内外研究状况综述

### (一) 国外研究状况

随着移动应用的发展, iOS 和 Android 已经成为了我们最为熟悉的两大平台,但是这两大平台都有自己独立的 SDK (Software Development Kit), 但正因为平台特有的 APIs 的存在, 开发者可以创造拥有丰富用户体验, 高效表现的客户端应用。但也正因如此, 跨平台移动应用的开发和维护变的复杂而昂贵, 并消耗了大量开发的时间和测试的时间, 以下是几大常用平台的对比:

表 1 从开发的角度看移动平台之间的差异

| OS                      | Programming Language | Development Environment        | Application Store     |
|-------------------------|----------------------|--------------------------------|-----------------------|
| Google's Android        | JAVA                 | Android Studio, Android SDK    | Play Store            |
| Apple's iOS             | Objective-C/Swift    | XCode                          | Appel-iTunes          |
| Microsoft Windows phone | Visual C#, C++       | Visual Studio                  | Window Phone Market   |
| RIM BlackBerry OS       | JAVA                 | BlackBerry Plug-in for Eclipse | BlackBerry Apps World |

由于人们对于应用的开发与配置速度要求的提高, 拥有与客户端应用一样的功能, 但是只使用单一的 SDK 工具并可以跨平台工作的移动应用开始出现<sup>[5]</sup>它允许开发者只使用一种语言, 只使用一种框架, 但是却可以被翻译到多种平台使用, 因为最下端的机器语言就是二进制语言, 所有语言的基础是相通的<sup>[6]</sup>但是, 框架在跨平台方法和范例中却有着显著的差异。如图: 跨平台方法的分类<sup>[7]</sup>但是自跨平台应用程序开发框架的出现, 多平台创建应用开始变的容易, 它减轻了开发者学习上的压力, 同时降低了成本并加快了产品上市的时间<sup>[8]</sup>, 但是对于图形密集型的游戏, 不得不承认跨平台应用还是无法完美的展现。

在探索跨平台应用的过程中, web 的方法开始被提出。Web 方法建立在远程服务器的基础之上, 其移动应用程序使用 HTML, CSS 和 JavaScript 实现, 通过标准协议 (如 HTTP) 提供服务, 并通过唯一的 URL (Uniform Resource Locator) 访问。总而



言之，web 移动应用就是将用户移动设备浏览器（如 Chrome，Firefox）作为其运行时的环境，并受益于移动平台浏览器的支持的移动优化网站。

在这种方法中，应用程序被实现为移动的单个优化网站。但是此优化必须考虑设备的不同屏幕尺寸以及其使用的原理。Web 移动应用程序的优势在于它们在所有平台上的浏览器中以类似的方式存在。因此，不需要移动应用程序更新。 Web 方法的缺点是访问设备的本机功能（例如通知系统，GPS，联系人列表等）是有限的。第二个缺点是通过从网络加载来呈现网页所花费的时间比本机移动用户界面的时间长。此外，Web 应用程序只能通过 URL 访问，并且无法在移动应用程序商店中随时可用。这会对方法的吸引力产生越来越小的影响，以下是 web 方法的图形解释<sup>[9]</sup>。

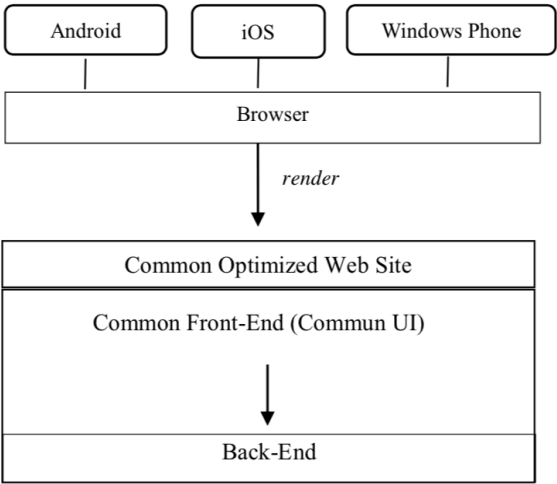


图 2 web 方法的图形解释

其中 JQuery Mobile<sup>[10]</sup>是最流行的框架，它提供了高品质的响应式网页，并可以与最流行的智能手机，平板和电脑兼容。Jquery Mobile 以 HTML5 为基础，jQuery 和 jQuery UI 为其提供技术支持，并拥有页面转化，触摸事件，和各种各样的工具提供支持。它的轻量级代码使用了渐进增强功能，并具有灵活，易于设计的优点。

随着人们对 web 端移动应用的不断摸索，渐进式 Web 应用程序（PWA）开始出现，渐进式 Web 应用程序是一种特殊类型的移动 Web 应用程序，它的功能逐步加强，支持低速和无网络状态下的使用，在后台处理功能，推送通知支持和安全性上做的也非常的出色。同时渐进式网页应用作为一项新颖的开发方式，承诺将 Web 技术的易用性和本机应用程序的多功能结合起来<sup>[11]</sup>。其与 web 站点式应用的不同也在于 PWA 提供了前者无法提供的附加功能和用户体验<sup>[12]</sup>。与其它移动网络应用程序类似，PWA 通过 HTTPS 远程服务器提供服务，并且可以通过浏览器作为初始访问的窗口（即不

需安装就能使用), 由用户决定是否安装 PWA, 并使用更多功能, 如全屏支持 (没有浏览器标签), 通过专用的缓存管理器进行缓存管理和消息推送等客户端应用所具有的功能。

从技术的角度来讲,PWA 是一个网页和一个建立在 JSON 基础上的元数据文件<sup>[13]</sup> (manifest 清单) 一个应用程序外壳和一个使用 JavaScript 编写的后台脚本 (Service worker 服务工厂)。服务工厂处理和代理所有的网络连接, 储存后台任务, 从而实现 PWA 的离线体验<sup>[14]</sup>。此外网站通过 HTTPS 提供服务并实现响应式设计<sup>[15]</sup>。

与传统的网页相比, 渐进式网页应用程序 (PWA) 多了更多的功能, 它可以像本地 app 一样线下使用, 并且可以向用户发送通知, 并且脱离了浏览器的限制可以将网页脱离出来以图标的形式在手机桌面上单独出现, 总而言之, 渐进式网页应用技术给予了网页移动 app 拥有的一切功能。

截止今天, PWA 提供的功能在很大程度上还是取决于浏览器的支持标准, 比如, 在最新的 Chrome 中 PWA 可以支持推送通知, 访问摄像头, 访问地理位置等功能, 但在不支持的浏览器中就变成了传统的网页不具有后续增强的功能。同时对于访问低级硬件功能和传感器也并不支持, 如访问联系人, 日历, 闹钟和短信, 电话和用户本机号码等<sup>[16]</sup>。

下表是对 Hybrid, Interpreted, PWA 三种移动应用开发方式的比较, 从安装所占内存, 启动时间, 开启应用时间这三方面进行比较, 其中渐进式网页应用程序所占的内存比以本地应用为基础的 interpreted app 小了约 157 倍, 比框架基础的 hybrid app 小了约 43 倍, 同时比 hybrid app 少用了 9 秒的时间从点击图标到显示 app, 而 interpreted app 用了将近 860ms 完成相同的工作。

表 2 Measurement-comparison of approaches

| Measure                                  | Hybrid   | Interpreted | PWA                      |
|--|----------|-------------|--------------------------|
| Size of installation                     | 4.53MB   | 16.39MB     | 104KB                    |
| Launch time                              | 860ms    | 246ms       | 230ms                    |
| Time from app-icon tap to toolbar render | 9242.1ms | 862ms       | (a) 3152ms<br>(b) 1319ms |

自 PWA 出现之后, 人们对 PWA 的好奇心开始增加, 但是不可否认的是这还是一项没有广泛推广的新兴技术。在 2017 年一月, 有人就 PWA 相关的搜索结果进行了

调查,但是搜索结果是否有限,在 Google Scholar 中只导出了两篇文章,用同样的搜索词条, Taylor&Francis 和 ScienceDirect 数据库一样返回零结果, ACM 电子图书馆返回了与 Google Scholar 相同的两条结果。同时对于与其相关的术语的搜索如 Service Worker,web,API 等,其搜索结果受不相关的结果污染严重<sup>[17]</sup>。在 2019 年 5 月,当我再次搜索时,相关的搜索结果已经非常庞大。

对于渐进式网页设计的应用范围,国外是非常的广泛,比如印度,由于印度的经济水平不高,人们的手机内存都不大,而且网速普遍较慢,这让 PWA 在印度成了最流行的新技术。而且 Blink (Chrome、Oprea、Samsung Internet 等)与 Gecko (Firefox)都已经实现了 PWA 所需的关键技术,并已经开始探寻更多的可能性。而 2018 年正是 PWA 的破局之年,目前,全球顶级的浏览器厂商, Google, Microsoft, Apple 已经全数宣布支持 PWA 技术,关于 PWA 的一切都在不断的推进中。

“Gartner has stated that 50% of all native apps will move to be progressive web apps by 2020.” Gartner 表示,到 2020 年,50% 的原生应用程序将成为渐进式网络应用程序,由此可见,渐进式网络应用程序正在以燎原之势快速的发展着。

## (二) 国内研究状况

在移动应用方面,随着互联网和智能手机的普及,移动应用程序的数量呈指数形式的增长。截止 2012 年 10 月份,苹果 App Store 和 Google play 移动应用数量均超过 70 万。App Store 平台应用下载总数超过 500 亿次, Google Play 平台应用下载总数超过 480 亿次<sup>[18]</sup>。由此可见,移动应用程序已经在我们生活中已经占据了非常重要的位置。

但是客户端的应用程序 (Native App) 会因为硬件本身的限制而受到很大的影响,比如移动设备内存的限制,在应用开启下电池工作的时长,可访问空间的大小等<sup>[19]</sup>,同时移动设备也有带宽低,计算能力弱,连入网络的能力差,连入网络时时间成本高等问题。同时本地应用也面临着平台不兼容的问题,以我们所熟知的安卓和苹果两大平台为例,移动应用的开发者如果想开发出移动应用并在两个平台下实现,那么针对安卓,苹果两大平台写出针对性的版本就很必要了。所以,让移动应用有跨平台实现的能力,将会大大的方便开发者的开发过程<sup>[20]</sup>。

其中跨平台的目的是在保持代码不变或是少数变化的情况下,将应用程序转移到其它平台并保持应用程序的功能不变。同时可以跨平台的编译框架会处理接口之间的

差异性,这样程序员在开发的时候就不用考虑接口的差异性而是直接使用统一接口就可以兼容多个不同的平台,这样可以极大的节省开发人员的精力和时间。

而 web 方法作为实现跨平台目的的方法之一,其由 HTML5, JavaScript 和 CSS 提供技术支持。其中 HTML5 用于界面和窗口部件的显示,并取代了传统客户端应用中 Android 和 iOS 的 native 部件,从而达到跨平台实现的效果。JavaScript 在跨平台实现中起到了核心作用,它与 HTML, CSS 一起合作完成界面显示,同时对后台的运作提供支持,如通过 Ajax 实现网络请求,实现数据缓存,与本地 API 交互。CSS 同 HTML5 一起完成界面实现,并对界面的布局和样式完成设计。因为本地功能如相机,短信,通讯录,地址访问等需要本地的 API 完成实现,所以 JavaScript 通过调用本地的 API 完成这些功能。



图 3 JavaScript/HTML5/CSS 跨平台架构图

同时与 web 相关的技术也对基于 web 技术的移动应用提供了非常多的帮助,如 jQuery, jQuery 是一个功能强大的 JavaScript 图书馆,通过 JQuery 可以轻松的实现 HTML5 的节点访问,修改,还有对事件和动画的处理。同时 jQuery 对网络请求进行了进一步的封装,使得用户可以轻松的访问网络请求和数据处理,由于 jQuery 的平台独立性,开发者也可以轻松的调用 jQuery 完成功能的实现。与此同时,完整的数据在内存中完成与其它数据的交互并独立与程序存在,这样有利于数据的双向转化。jQuery Mobile 是基于 HTML5 的一个跨平台的 UI 框架,可以应用于平板和手机等移

移动端，是移动应用程序开发的一个 UI 数据库。其提供了许多界面控制工具如 Button, CheckBox, ListView, Dialog 等等，与 jQuery 一样这些工具也拥有跨平台的属性，因此程序代码可以在多个平台上成功完成共享<sup>[21]</sup>。

就 web 应用程序的开发而言，这种方法的难度不高，门槛也很低，只要掌握了 HTML5, JavaScript 和 CSS, 并且熟悉布局和数据请求的实现就可以进行 App 的开发。对于 web 界面开发来说，界面由 HTML5 和 CSS 完成，通过 JavaScript 完成对用户操作的响应，这和普通网页的开发相似，所以开发的效率很高。在性能方面，web 型应用还不能很好的展现超大型应用，很容易造成页面冻结，对用户操作的响应也会变得缓慢<sup>[22]</sup>。

对于像植物图鉴 app 这样知识型应用（帮助读者随时随地获取信息）本身的性质如公益性，免费的向广大读者提供信息，随时随地的提供服务，个性化，用户可以根据自己的需求选择所需的阅读材料不需要一次性拥有广泛的材料等性质<sup>[23]</sup>，注定了它的用户量和活跃度不会特别高，就比如没有人每天都会用 Wikipedia 但一定有人每天都会上微信一样，知识型的应用在腾讯 QQ，微博，微信这类超大型应用的竞争中完全没有优势。同时在传统的应用商店运营模式下，“长尾效应”<sup>[24]</sup>也是知识型应用用户量不高的一大促因，即大部分的流量只会流向头部，也就是那些超大型的应用，而接触不到像知识型应用这种小而精的应用。这也使知识型应用只能在应用商店中成为那些无人问津的“僵尸应用”<sup>[25]</sup>而使用 web 形式的 app，正好可以避开 app 商店中强大的竞争，给小而精应用更多的生存空间。

同时应用 WEB 技术对于像植物图鉴 app 这样的知识型移动应用是一个非常好的选择。首先知识型的应用不属于复杂程序，它没有也不需要非常多的功能，这也就没有了充分利用平台优势的意义。同时，知识型的应用效果就是让人们可以随时随地获取知识，在移动网络普及的情况下，WEB 型应用就可以做到这一点，再加上 WEB 型应用可以很好的解决平台局限性的问题，因为 web 的应用程序在标准的浏览器中运行，所以完美的跳过了硬件和操作系统对应用程序的限制，只要用户使用浏览器就可以随时随地的使用 web 浏览信息。再加上 web 应用本质上是 client-server 型应用也就是说在应用需要更新的时候只有服务器端需要更新，客户端是没有更新烦恼的，这也给了知识型应用这种小众应用一些竞争优势。建立在知识型应用需要对版权的保护，把数据放在服务区端而不是客户端也提供了更高的安全性，开发者不再需要担心

因为客户端的服务器被攻击而导致的数据流失<sup>[26]</sup>。

作为 web 应用中最新兴的技术，利用 PWA 来完成知识性应用是一项非常好的选择，它支持网页端和移动端同时工作，用户如果使用频率小网页端完全可以满足需求，如果用的多使用渐进式 web 应用既可以在任何设备上及时的吸收知识，有免去了下载所需的大量时间。

但是关于这项技术的内容在中国还没有开始流行，截止 2019 年 5 月 7 日，以百度文库搜索为例，搜索 PWA 相关论文只出现了一篇，基于 WebRTC 和 PWA 的视频互动直播系统<sup>[27]</sup>，但是其中代码实现为零，也就是纯理论无法实现，而其它的多以 ppt 形式的文案为主，同时其在中国知网的搜索中资料就更少，搜索关键字 PWA，搜索出相同的基于 WebRTC 和 PWA 的视频互动直播系统，和基于 Progressive Web Apps 的兼职服务系统，这篇文章中只有两段但是时效性太差是非常老旧的编程方法<sup>[28]</sup>。但在 google 官网，2019 年 4 月 30 日，全新的 PWA 教材开始使用，其中使用了更高效更清洁的代码。除了这两篇外其它结果受非相关结果污染严重，出现大量化学类论文结果和物理类论文结果，搜索中文名称渐进式 web 应用出现大量的医学类结果，首页真正相关结果为零，使用全称 Progressive Web App 进行英文搜索结果为零，中文搜索结果不相关。所以可见 PWA 作为一项在多个国家兴起的新兴技术并没有受到重视，人们对其开展的研究寥寥无几。所以 PWA 作为一项独具创新的项目在中国的发展才刚刚开始

### 三. 植物图鉴移动应用程序开发

#### （一）应用程序开发过程

而我将使用渐进式网页应用技术来完成我植物图鉴移动应用的设计工作。植物图鉴移动应用是一个典型的知识型应用，它不没有复杂的人机交互功能，没有过多的功能键，也不需要过多的访问本机中的硬件资源如物理地址，摄像头等，它只是一个为人们提供养花草知识的一个小型应用，而我做这个应用的目的也不是为了让人们真正的从其中了解相关的知识而是用这个小应用来让大家了解渐进式网页应用技术的本质和魅力，同时展现出其重要功能的完成和相关的代码展现。

其中我将完成 1) app 的展示设计，使用 App Manifest（应用清单）完成本地移动应用点击桌面图标就能进入程序的功能，2) 简单的离线功能的实现，在这一步中，

如果 app 遇到了无网的环境将展现自定义离线页面而不是浏览器默认界面，3) 安装功能，让 app 真正的脱离浏览器的限制以与客户端 app 一样的形式出现在移动设备的 Home 界面上而不是隐藏于相关的浏览器应用中。

## （二）系统的总体架构：

系统的主体方面，我使用 HTML5 完成，先确定卡片区块，通过<div>来帮助完成卡片的区域设定，然后是确定卡片内容，就用户的需求而言，植物图鉴 app 需要清晰度足够高并且拥有植物独特特征的图片并配有其名称，再就是对于如何养育的指导给出相应的信息，这些就是每一个独立卡片中内容的定义。

在就是对卡片展示风格的定义，在这一方面我使用 CSS 来完成相关定义，第一，我选择使用白底卡片，因为白底与黑底是反差最大的两种颜色，但考虑到我找的植物图片均是亮色系所以我选择使用白底来给大家一种清新和自然的感觉，所以没有选择黑色这种压抑的色系。再就是对与卡片中元素位置的确定，植物图片我选择使用左悬浮的方法完成，这样可以完美的与文字区分，从而不会产生拥挤的感觉，再来是文字，文字的位置我选择使用 margin 来与图片间隔出适当的距离，并用 Helvetica Neue 形字体对 h2 元素也就是卡片中植物的名称进行加粗，最后正文部分我选用 Arial 字体来书写，以下是简单的代码实现，其中 img 指代图片元素，p 指代正文内容，和完成后的成品展现：

```
<div class = picture>
```

```
    
```

```
    <h2>Asparagus Fern</h2>
```

```
    <p>
```

```
        Asparagus setaceus adapts to both bright spots and darker corners. Keep the soil moist  
and it'll thrive.
```

```
    </p>
```

```
</div>
```

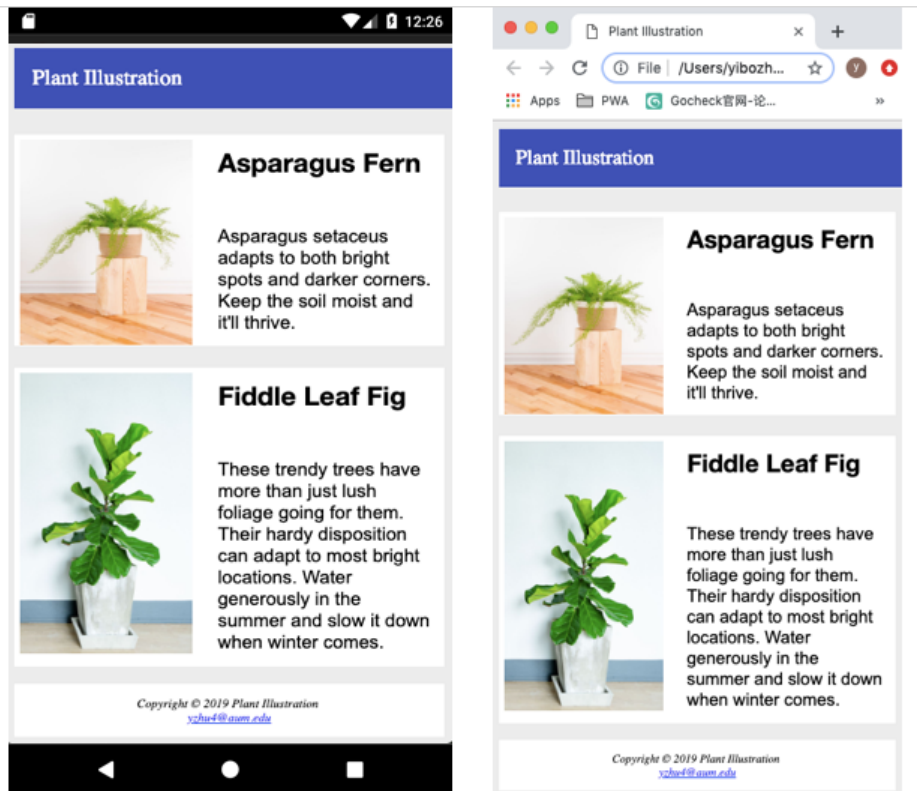


图 4 植物图鉴 app 完成

### （三）渐进功能的实现：

我的应用将会使用（1）反应式设计（reponsive design），因此它将可以在 PC 端和移动端同时工作。（2）为了使之快速，我将使用 service worker(服务工厂)预先运行和存储所需的应用程序资源（HTML，CSS，JavaScript，图像），并在运行的时候缓存植物数据以提高性能。（3）使用 WEB 应用程序清单和 beforeinstallprompt event 去通知用户它是可以安装的。

#### 1.App Manifest（应用清单）的实现

应用清单的目的是提供应用开发者修改展示界面的能力，这些修改包括图标的路径，程序的名称，启动画面的展示等，简而言之 Web 应用清单是一个简单的 JSON 文件，它告诉浏览器，移动设备或桌面安装时应用应该如何表现即用于修改程序的行为和样式。

应用清单将提供应用信息如名字，作者，图标，和描述等，并将它们存储在 JSON 文件中，开发者通过它将有能力控制应用的展示方式，如定义应用程序图标，启动应用程序（独立，全屏，浏览器等），同时使 web 像 App 一样展现，Manifest 在 植物图鉴 app 中的作用有：



- a.告诉浏览器将 app 在一个单独的窗口打开（display）
- b.定义首次启动应用时打开的页面（start\_url）
- c.在 Dock 或 app 启动器上定义应用程序的外观（short\_name,icon）
- d.创建启动画面（name,icons,colors）
- e.告诉浏览器以横向或纵向模式打开窗口（orientation）

```
{
  "name":"Plants",
  "short_name":"Plants",
  "icons":[{
    "src":"/images/icons/icon-128x128.png",
    "sizes":"128x128",
    "type":"image/png"
  },{
    "src": "/images/icons/icon-512x512.png",
    "sizes": "512x512",
    "type": "image/png"
  }],
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#3E4EB8",
  "theme_color": "#2F3BA2"
}
```

以上是应用清单的实现代码，因为硬件的屏幕大小不一所以为不同的屏幕匹配不同的 icon 与之对应是非常必须的，在此我只实现了两个 icon 的设置，接下来的设置可以以此为例，同时需要告诉浏览器应用清单，通过加<link rel= “manifest”href= “/manifest.json”>到 index.html 中来实现。因为 ios 的 Safari 现在并不支持网页形式的应用清单，所以为其编写的传统 meta tags 如下：

```
<!-- Add iOS meta tags and icons -->
```

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

```
<meta name="apple-mobile-web-app-title" content="Plants PWA">
```

```
<link rel="apple-touch-icon" href="/images/icons/icon-152x152.png">
```

## 2.service worker（服务工厂）的实现

服务工厂负责渐进式网页应用的核心功能，如果没有服务工厂的支持，PWA 应用将无法在浏览器中正常的工作。它在用户首次访问时被注册并由一个 JavaScript 文件组成，该文件包括了运行逻辑和缓存周期的一套循环过程。服务工厂也可用于处理任务，如后台同步，数据和应用外壳的缓存，以及拦截网络请求。处于安全的原因，服务工厂需要使用在浏览器中注册的 HTTPS 来处理相应的事件，因为服务工厂的能力太过强大，它可以劫持相应的网络连接，过滤相关的响应，还可以伪造虚假的回复。

在这一步中我将提供简单的线下服务，因为用户希望在应用程序离线的时候始终都有基础的页面展现，所以此线下服务将从一开始的简单离线页面到具有可阅读已缓存的数据再到功能完全的脱机体验并在网络连接恢复的时候完全同步的一系列应用服务。同时提供 Service Worker（服务工厂）的应用就可以进一步被考虑为拥有 progressive enhancement(渐进式增强功能)的应用了。当服务工厂在支持的浏览器上使用时，它将缓存 App shell（应用程序外壳）和应用程序数据，以便在网络不可用时用户也可以正常使用应用，但对于不支持的浏览器，服务工厂就不会调用脱机代码，用户也只能获得最基础的体验，而且服务工厂的消耗是很小的，也不会影响旧浏览器的正常使用。这也就是所谓的向上支持但是向下不兼容但是不影响。而安装服务工厂的第一步就是注册，相关代码实现如下（储存于根目录 index.html 下）：

```
//Register service worker
if ('serviceWorker' in navigator) {
  window.addEventListener('load', () => {
    navigator.serviceWorker.register('/service-worker.js')
      .then((reg) => {
        console.log('Service worker registered.', reg);
      });
  });
}
```

将服务工厂储存于根目录下是因为这是设置服务工厂操作范围的最简单的方法，服务工厂的操作范围是指其将控制那些文件，即服务工厂将从那条路径来拦截请求。默认范围是服务工厂所在位置和其下所有的子目录，因此将 `service-worker.js` 储存在根目录中，服务工厂就能控制此应用的所有请求。

再来是定义服务工厂在离线的情况下将储存的离线页面，它将在没有网络连接的情况下展示，以下是主要代码和展示内容（存于 `/offline.html`）：

```
<body>

  <header class="header">

    <h1>

      Plant illustration

      <a href="https://darksky.net/poweredby/" class="powered-by"></a>

    </h1>

  </header>

  <main class="main">

    <div class="plant-card">

      <div>

        Oops, you appear to be offline, this app requires an internet
        connection.

      </div>

    </div>

  </main>

</body>
```

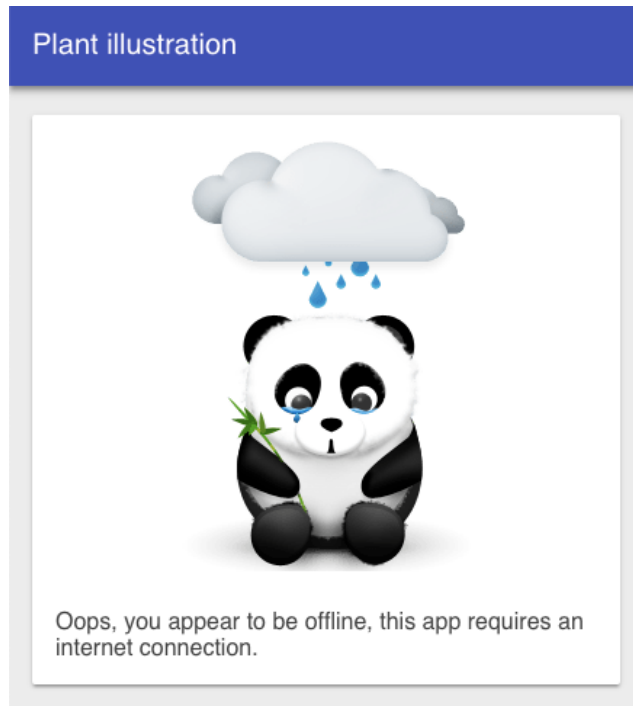


图 5 离线页面完成

接下来就是将“/offline.html”加在 service-worker.js 中：

```
//Update cache names any time any of the cached files change.
```

```
const FILES_TO_CACHE = [
```

```
  '/offline.html',
```

```
];
```

并更新 install event 来告诉服务工厂需要事先存储离线页面：

```
// Precache static resources here.
```

```
evt.waitUntil(
```

```
  caches.open(CACHE_NAME).then((cache) => {
```

```
    console.log('[ServiceWorker] Pre-caching offline page');
```

```
    return cache.addAll(FILES_TO_CACHE);
```

```
  })
```

```
);
```

其中 `cache_name` 帮助我们更新其中的信息但是不影响其他信息。

最后，就是处理 `fetch events`，使用 `network, falling back to cache strategy`，也就是服务工厂将抓取来自网络的数据，如果失败，它将使用离线页面。代码实现如下，其中 `evt.respondWith()` 用来阻止浏览器的默认操作并告诉浏览器由用户自定义处理：

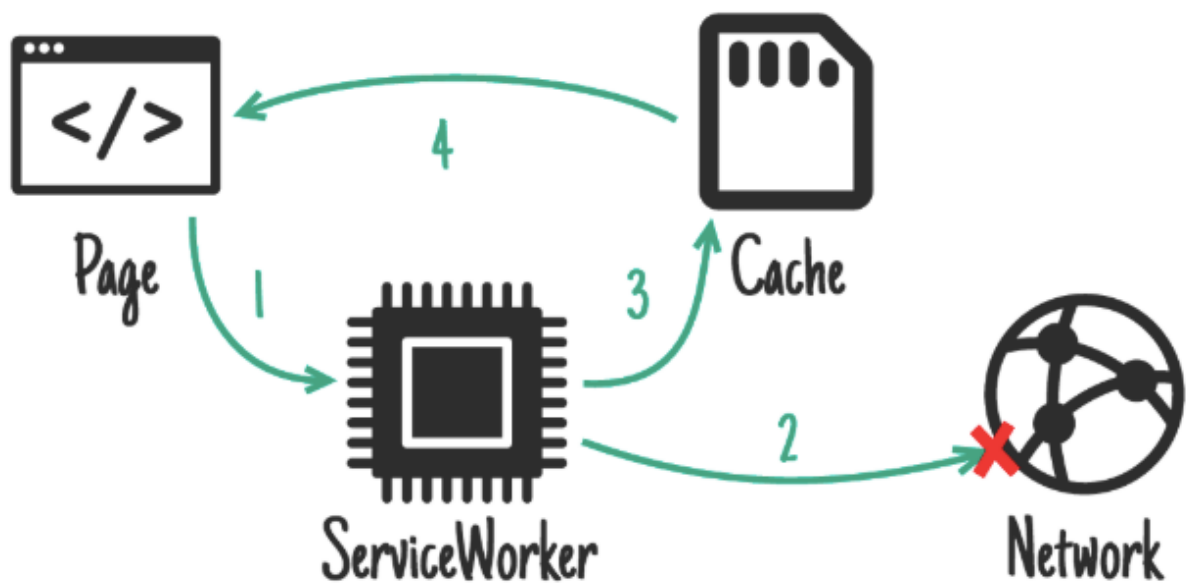


图 6 服务工厂工作原理

```

// Add fetch event handler here.
if (evt.request.mode !== 'navigate') {
  // Not a page navigation, bail.
  return;
}
evt.respondWith(
  fetch(evt.request)
    .catch(() => {
      return caches.open(CACHE_NAME)
        .then((cache) => {
          return cache.match('offline.html');
        });
    })
);

```

### 3. 安装功能的实现

接下来就是添加安装的功能了，当一个渐进式移动应用被安装在主页面时，它的外观和行为将和其它的客户端应用完全一致，它将和其它移动应用储存在同一个地方，并且作为一个独立的移动应用运行，不需要地址栏和其它的浏览器支持，也就是

成为像客户端应用一样的高性能应用。同时添加安装功能可以通过谷歌默认的三点上下文菜单来安装渐进式 Web 应用程序，也可以自定义按钮或其它 UI 组件，来提供应用安装功能。

第一步就是将 `install.js` 添加到主页面 `index.html` 中

```
<!--Add the install script-->
```

```
<script src = "/scripts/install.js"></script>
```

如果条件支持渐进式移动应用的功能，`beforeinstallprompt` 事件将被触发，指示此应用可以被安装到主屏幕，然后允许用户的安装。在 `saveBeforeInstallPromptEvent` 功能中，一个参考目录将被储存在 `beforeinstallprompt` 事件下，所以在我们呼叫 `prompt()` 功能的时候，安装按钮将显示在用户界面上，`prompt()` 将显示一个对话框，询问是否添加 app 到主屏幕。

```
// Add event listener for beforeinstallprompt event
```

```
window.addEventListener('beforeinstallprompt', saveBeforeInstallPromptEvent);
```

因为当用户点击安装按钮的时候，在 `beforeinstallprompt` 事件中的 `.prompt()` 将被触发，然后安装按钮需要被隐藏因为移动应用只需要一次安装，所以安装按钮是一个一次性按钮。

```
// Add code to save event & show the install button.
```

```
deferredInstallPrompt = evt;
```

```
installButton.removeAttribute('hidden');
```

```
// Add code show install prompt & hide the install button.
```

```
deferredInstallPrompt.prompt();
```

```
// Hide the install button, it can't be called twice.
```

```
evt.srcElement.setAttribute('hidden', true);
```

接下来就可以通过 `userChoice` 来监听用户对安装对话框的回应。

```
// Log user response to prompt.
```

```
deferredInstallPrompt.userChoice
```

```
.then((choice) => {
```

```
  if (choice.outcome === 'accepted') {
```

```
    console.log('User accepted the A2HS prompt', choice);
```

```

    } else {

        console.log('User dismissed the A2HS prompt', choice);

    }

    deferredInstallPrompt = null;

});

```

最后因为已缓存的文件被进行了更改，所以更新 `service-worker.js` 文件中的 `CACHE_NAME`，下图为用户点击安装按钮后系统自动显示的安装提示窗口。

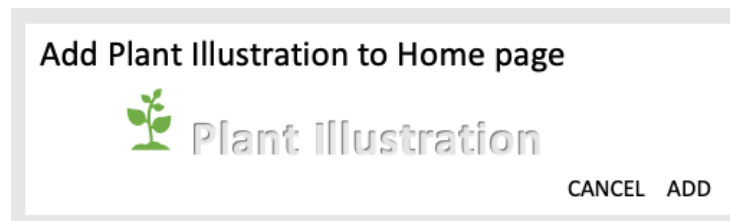


图 7 安装提示窗口

#### 四．总结与展望

作为移动应用开发技术的新星，PWA 在中国还没有得到它应该得到的重视，关于它的研究还很稀少，但是科技发烧友们已经开始在各大论坛开始了关于其技术的讨论，也正在投入力量到渐进式网页应用程序和其相关的学习资料的开发过程中。但是相对缺乏的学术资料意味着和美国的知识差距巨大，但也反向证明了中国的研究潜力巨大。本文旨在通过介绍 PWA 相关概念和技术，比较其性能，展现其核心代码，并实现一个小型而简单的植物图鉴 APP 来全面展示这项技术，并希望可以提高学术届对 PWA 这一项新技术的认识。最后我以在谷歌 I/O 大会上的结束语做结：“We want everything that ends up on the home screen to be competitive with native apps. We want to make the web a first-class part of the operating system in the user’s mind”。

移动应用程序的开发是一个不断发展和动态变化的过程，其中很多的机遇和挑战来自于使用 web 技术开发移动应用，很难想象 web 的初衷与移动应用的需求有着惊人的相似，它们都追求着以最简单的方式让所有人随时随地获得想得到的信息和体验。但是做为一个新兴的技术不可否认的是它还有着许多的缺陷和不足，就本文而言，其对 PWA 功能的实现还是非常的有限，比如对离线功能的实现是可以从一开始的简单离线页面，到具有可阅读已缓存的数据，再到功能完全的脱机体验并在网络连接恢

复的时候完全同步这一系列应用服务的，但是本文只实现了简单的离线页面这一单一功能，接下来关于服务工厂的复杂功能还是需要接下来的研究者不断的去探索，同样对于正文中未提及的技术如消息推送，对本地硬件访问的实现也是很有意义的技术。

以下是在技术，经济，社会影响方面对 PWA 进一步研究提出的建议：

- 在社会经济的方面，研究对于发展中国家，移动数据费用对下载应用市场中的程序和渐进式网页应用程序的影响。
- 研究“添加到主屏幕”与通过应用程序市场安装对应用程序的影响。用户是否更愿意安装 PWA，因为可以事先测试它？临时使用和安装最终会合并吗？
- 应用程序市场拥有相应的规则束缚并提供一定程度的安全性，对于缺乏这些潜在限制的 PWA 应用程序，这会不会成为影响用户对安装 PWA 应用的选择的原因之一？
- PWA 技术是否引入了新的安全和隐私风险？
- 在研究渐进式网络应用程序的可能销售模式的过程中，这些模型会扰乱已建立的应用程序市场秩序吗？



## 致谢

首先，我要感谢我的导师彭瑞卿副教授，感谢他提出的每一个宝贵的意见，并在论文完成的全程对我的帮助和鞭策，再来我要感谢院系各位老师和领导，感谢他们对我们这一批在国外留学的学子的关心和帮助，同时我也要感谢我们同批出国的同学们，是你们的鼓励让我在短暂的时间下顶着压力完成了我的毕业论文，最后我要感谢我的家人，是他们的关心和提醒让我可以按时的完成我的毕业论文。最后的最后，感谢吴磊老师对我选题提供的灵感，并在全程给我信心让我相信就算是本科生的毕业论文也可以对世界级的先进技术发起探索！

## 参考文献

- [1] CNNIC 中国互联网发展统计报告（第 41 次报告）.2018.01
- [2] Taylor, D.G., Voelker, T.A. and Pentina, I. (2011) Mobile Application Adoption by Young Adults: A Social Network Perspective. *International Journal of Mobile Marketing*, 6, 60-70.
- [3] Gupta, S. (2013) For Mobile Devices, Think Apps Not Ads. *Harvard Business Review*, 71-75.
- [4] Mobile APPs and Global Markets, from Kungpo Tao, Paulette Edmunds, Department of Business and Economics, Elizabeth City State University, Elizabeth City, USA, June 12, 2018
- [5] Paulo R. M. de Andrade, Adriano B. Albuquerque. “CROSS PLATFORM APP A COMPARATIVE STUDY”. *International Journal of Computer Science & Information Technology (IJCSIT)* Vol 7, No 1, February 2015
- [6] . Native,Web or Hybrid Mobile-app Development. White paper, IBM Corporation, April 2012.Document Number:WSW14182USEN
- [7] T. A. Majchrzak, J. Ernsting, and H. Kuchen, “Achieving business practicability of model-driven cross-platform apps,” *OJIS*, vol. 2, no. 2, pp. 3–14, 2015.
- [8] A. Ribeiro and A. R. da Silva, “Survey on cross-platforms and languages for mobile apps,” in *Proc. 8th QUATIC*. IEEE Computer Society, 2012, pp. 255–260.
- [9] Cross platform approach for mobile application development:a survey,DOI:10.1109/IT40D.2016.7479278
- [10] JQuery Mobile.Last access:December 2015.<https://jquerymobile.com/>
- [11] A. Biørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, “Pro- gressive web apps: The possible web-native unifier for smo- bile development,” in *Proc. 13thWEBIST*. SciTePress, 2017, pp. 344–351.
- [12] P.LePage,“Yourfirstprogressivewebapp,”2017,<https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/>.
- [13] M. Guant and P. Kinlan, “The web app manifest,” 2017,<https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/>.
- [14] M. Gaunt, “Service workers: an introduction,” 2017, <https://developers.google.com/web/fundamentals/getting-started/primers/service-workers>.
- [15] M. Nebeling and M. C. Norrie, “Responsive design and development: Methods, technologies and current issues,” in *Proc. ICWE*. Springer, 2013, pp. 510–513.
- [16] Beyond Native Apps: Web Technologies to the Rescue! Ivano Malavolta Vrije University Amsterdam,The Netherlands,2016 ACM.978-1-4503-4643-6/16/10
- [17] . Biørn-Hansen, A., Majchrzak, T. and Grønli, T-M. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. DOI: 10.5220/0006353703440351. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 344-351. ISBN: 978-989-758-246-2

- [18] 张栋栋, 徐锋, 一个移动应用个性化集成框架的研究及其在 Android 平台的实现, 计算机科学, 2014.11 月, 63
- [19] 面对移动云计算弹性应用的安全模型, 徐光侠, 陈蜀宇, 计算机应用, 2011 年 4 月 953
- [20] 面向移动 WEB 的微手势识别技术及系统实现, 张美玉, 陈嘉城, 张伟, 2016, 6 (1), 81-89  
数据挖掘
- [21] 基于 JQuery 框架的 Web 移动端快速开发技术研究与应用 吴国辰, 河源技师学院 2016 年 5 月
- [22] Android 和 iOS 跨平台编程的研究, 彭广军, 上海交通大学电子信息与电气工程学院, 2015 年 5 月
- [23] 移动图书馆的研发与实现 顾国庆 南昌大学 2014 年 12 月
- [24] 吴爱琼. 移动互联网平台中的移动图书馆服务. 图书馆学刊, 2012 (1) :80-83
- [25] 移动互联网下图书馆轻应用服务模式, 张毅, 情报资料工作, 2015, 36 (4) :75-78
- [26] 基于移动 Web 的数字图书馆设计与实现 肖燕红 吉林大学 2015 年 6 月
- [27] 基于 WebRTC 和 PWA 的视频互动直播系统 彭永超, 费璟昊 1. 湘潭大学, 湖南 湘潭 411105; 2. 北京斯邦医疗科技有限公司, 北京 102200 2017 年 2 月, 电脑编程技巧与维护
- [28] 基于 Progressive Web Apps 的兼职服务系统 韦王强, 方姚, 王玉, 文章编号:1005-1228(2017)06-0043-05