

云计算应用开发课程

应用项目计划书

项目名称:	基于云计算的自助旅游详细方案订制
队长姓名:	龚睿奇
队长电话:	13631239412 / 659412

团队成员	姓名	学号
	龚睿奇	12353048
	乔丹	12353???
	杨奇标	12353???

1 应用介绍

1.1 背景介绍

如今，人们的精神生活日趋丰富，逢年过节外出旅游是常有的事情。对于学生群体及上班一族而言，趁着周末在城市的周边或郊外进行短途旅行，亦很常见。随着人们对旅行的要求不断提高，由传统旅行社提供的传统意义上的“跟团”旅游服务，其缺点和不足日趋显现，已经不能满足一部分人的需求。因此，“自助游”这种旅游方式，即由自己来定制旅行的全过程，包括要去的景点，日程的安排，交通方式的选择，食宿的安排等应运而生。“自助游”的这种个性化，全定制的旅游方式十分地诱人，因此也赢得了广泛的欢迎。但是其缺点也很突出，概括而言就是一个词——“麻烦”：一切的饮食、住宿、交通等细节，都需要由自己来安排。这对一些没有接触过“自助游”的人来说，还是相当有门槛的。人们不仅仅要消耗额外的时间和精力去规划行程，而且在规划行程的过程中，很可能使用了错误或过时的信息，造成规划失误，以至于最终旅途不顺，甚至造成财产损失。正是“自助游”的这些缺点所带来的风险和不确定性，导致还有相当一部分人对其心存芥蒂。

经过讨论和分析，我们发现，人们在策划旅行的时候，真正需要去关心的，仅仅是那些要参观的景点，以及那些要进行的旅行体验；而至于除此以外的其他方面，例如交通，饮食及住宿等细节¹，都不属于旅行真正有意义的部分。而当一个用户选择了“自助游”，并且在规划自己的旅游行程时，他关心得更多的，却是这些不重要的“细枝末节”；并且，上述的“自助游”的风险和不确定性，也大多出自于此。鉴于此，我们认为，这其中存在着不合理性。而我们的这个应用，正是要去解决这不合理之处。

1.2 功能介绍

我们这个应用的目的，就是为用户订制完备及详细的“自助游”旅行方案，省去用户处理旅途细节的忧虑及烦恼。用户在使用这款应用，来规划自己的“自助游”方案时，只需要提交以下信息：

- 希望前往的地点（城市）；
- 有兴趣的景点或体验项目；
- 旅途时间，预算等少量其它信息，

我们的应用程序就可以根据用户提交的要求，为用户制定一份完整且详细的“自助游”方案。该方案具体将包括以下内容：

¹需要说明的是，很多时候，“食、住、行”三者本身也是旅行的意义所在。但这部分，我们将其归纳到前述的“旅行体验”中，此处所指的是除这些以外的部分。

- 旅行日程（哪一天参观哪些景点）；
- 食宿安排（用餐地点，酒店的位置）；
- 交通规划（交通方式及各段交通耗时）；
- 注意事项等其他信息，

并且将以顺序表或流程图的形式，直观地向用户展示我们的推荐方案。当然，用户拥有最终决定权，在我们呈现了推荐方案后，用户可以继续根据自己的需要来修改这份方案，直到自己满意，并做出最终的决定。方案被制定好以后，相关文件会在云端及用户本地分别保存，不仅增加了设备无关性，也防止在旅行途中由于上不了网而无法随时查看。

1.3 现有应用

必须承认，如今市面上已经有不少旅游相关的软件、手机应用及网站。对这些应用，我们可以大体上将它们划分为以下几个类别：有的提供旅游地点的信息，有的可以预定酒店，有的分享游记和旅行经历……【……………】

1.4 创新之处

上面列举了不少旅游相关的网站及应用。而总体而言，现有的应用，更多的是停留在信息传递方面，也就是说，仅仅是把大量的信息提供给用户。我们并没有发现一款软件或者应用，可以主动地为旅行者制定旅行方案提供直接的帮助。【……………】

2 开发方案

2.1 实施方案

在阐述我们的实施方案之前，有必要简要说明一下我们预想的这个应用的使用流程。在阐述完使用流程后，我们将结合使用流程，来讲解这个应用的具体实施方案。

首先，这个应用大体的使用流程列举如下：

1. 用户选定旅游目的地，具体到城市名，并且提供旅游预算，旅行时间等信息；
2. 应用展示该城市内及周边的景点，以及景点相关的信息，供用户挑选；
3. 根据用户的选择，为用户规划路线，安排行程及食宿，向用户呈现一份推荐方案；
4. 用户对方案进行修改后，保存。

其次，按照我们的设想，我们的应用将会由两个部分组成：用户手机中的应用软件（客户端），以及架设在云端的服务器。其中，正如大多数的云应用一样，本地的客户端，即用户手机中的应用并不进行太多的运算和操作，仅仅是为用户提供一个图形界面，收集用户的输入信息，并将这些信息上传到云端，由云端进行处理。云端处理完毕得出结果以后，将其发回给本地的客户端。本地的客户端对传回的数据进行一定的图形渲染，并最终将图形化的形式呈现给用户。因此，结合上述的使用流程，具体的应用整体实现方式如下：

建立云端数据库

这是搭建本应用的第一个步骤。我们认识到，如今已经有不少旅游相关的网站和信息提供者。因此，我们不需要生成我们自己的数据，只需要将已有应用的数据收集并整合起来，例如借助百度地图的API搜集地理信息，从大众点评网收集对餐厅和酒店的评价，从旅游地的当地政府部门网站中收集景点及政策的相关信息等。云端的一些优点，例如24小时不关机，网络带宽大，廉价且弹性存储空间等，就可以在这里被运用：我们可以将“爬虫”程序部署在云端，以比较优良的网络带宽，连续不断地收集并更新云端的数据，然后将这些数据，放到云端的数据库管理系统（DBMS）中进行维护和管理。对于“爬虫”程序，网络上已经有不少现有的程序可供修改和使用，我们设想的是使用Python语言来实现，而DBMS则使用MySQL。

前期交互

在手机App中保存一份中国城市的列表，用户在UI界面中选定其中一

个城市，并设置一些附加信息。用户输入完毕后，App将用户输入的信息上传到云端，云端根据一些内置数据及由其他地图应用获取的数据，收集选定城市周边的景点，将这些景点信息回传给用户。用户在手机App中选择好感兴趣的景点后，App向云端发送少量数据，指定这些景点。

决定景点的优先级

由于各个景点之间存在差异，为每个景点都分配同样多的时间，显然不是个好主意。因此云端先根据一套事先做好的，内置的标准，对这些景点做出区分并排序，排名靠前的景点，需要被安排比较多的时间，并且对它的行程要被安排得比较早。当然，在这个排序的过程中，用户对各个景点的偏好也应该被考虑。我们可以让用户在选择景点的时候顺便附上感兴趣的程度。

景点日程安排

包括各个景点的游览时间，以及前后景点之间的交通线路规划。在旅行的过程中，游客自然不希望在路途上花费大量的时间。这意味着，在安排日程的时候，我们需要对用户指定的景点按照它们的地理位置进行聚类，相互邻近的景点可以被连续地安排在一起，这样就可以减少用户耗费在路途上的时间。

食宿安排

安排好了景点的日程之后，就可以根据若干景点的地理位置，搜索其附近的餐饮及住宿服务。在搜索餐厅和旅馆的时候，也不需要对所有景点的周边都进行搜索，而只需要在那些安排于用餐时间前后游览的景点周边进行搜索。这么做，不仅减少了搜索量，降低了运算量，而且还有利于用户获得更人性化的旅游方案。这一步以及前两个步骤，我们计算使用C/C++程序，结合MySQL对C语言的API接口来实现。

整理并回传方案

至此，一份推荐给用户的方案就基本上在云端制作完成了。这份方案预计会以文本文件的形式被生成出来，云端需要将这份推荐方案回传到用户的手机App上，而手机上的App则读取这份文本文件，通过编排和渲染，以用户友好的方式将我们推荐的方案完整地呈现出来。数据回传的过程，我们预计使用云计算实验课上介绍的Python语言的webpy库，或者也可以用C语言程序创建一个Socket来实现。

用户修改并确认

用户可以细致地浏览这份方案，并且修改其中的任何细节。在用户修改完毕点击确认之后，如果原先的方案有被修改过，那么修改过后的方案文件会被上传到云端，同时也会在用户的手机中保留一份副本，方便用户快速地查看和浏览。

最后就是云平台的选择。综上所述，我们应用的实现，在云端要运用到Python，MySQL以及C/C++程序，并且还有一个用Java+XML实现的Android应用程序。可见，我们的云端必须是一台支持多种语言，而且功能相对完备的虚拟机。在选择云服务提供商的时候，我们必须选择那些可以在云上架设虚拟机的服务商。在云计算实验课上，TA们向我们展示了一些公司提供的云服务，其中最符合我们需求的，无疑是微软的Windows Azure。它提供的是一台配置可供选择的云端虚拟机，我们可以在上面做一切能够在本地PC上做的事情，例如运行操作系统，安装软件，配置开发环境等等。在Windows Azure上不仅可以选择虚拟机器的配置，还可以选择要安装的操作系统。根据我们的需求，我们认为Ubuntu 14.04LTS操作系统是我们的首选，因为它对上述三种语言都有很好的开发以及运行环境支持，并且对网络和内存的管理都比较好。

2.2 可行性分析

【.....】

2.3 关键技术问题

我们这个应用开发起来，还是有一定难度的，需要解决不少技术上的问题。下面将对已经考虑到的技术问题进行阐述。

收集网络上已有的信息，整理组成云端数据库

云端是一台24小时运行的虚拟机，它可以不间断地运行，收集大量的信息。云端需要收集的信息，大概分为以下几个类别：

1. 旅游网站提供的景点信息；
2. 地图网站提供的地理坐标及周边环境信息；
3. 点评类网站提供的客户评价信息；
4. 景点官方网站或地方政府网站的法律及旅游政策信息。

要收集网络数据，则离不开所谓“爬虫”程序。我们预计在网络上应该会有不少现成的“爬虫程序”的Python源代码，只要下载下来并做一定的修改，应该就可以使用了。因此，“爬虫”程序的问题不难解决。

不难想象，所谓的“爬虫”程序，就是以信息量大的网页（例如门户网站首页）为起点，以各个网页中存在的url链接为对象，使用搜索算法逐个访问每个页面，并获取上面的信息。可以想象，被“爬虫”程序收集下来的数据是原始的html文本，还夹杂着用网页脚本语言实现的例如动态广告及图片等冗余信息。我们必须通过一些软件及

算法，对收集回来的原始数据进行处理，将其中那些冗余的数据去除掉，提取出关键的、有用的信息。由于我们小组的成员没有做过数据挖掘方面的工作，因此提取有效信息这部分的工作，对我们来说是很具挑战的。

对景点优劣进行排序的方案和标准

这部分，是整个整个应用最关键的步骤，其地位可谓“承上启下”：承接上述收集信息的过程，开启下面对信息的处理。

景点的优劣，本身就是一个说不清，道不明的东西，其中夹杂着许多个人的兴趣与审美等感性因素。其后果就是，很难通过一个标准化的量化指标去衡量一个景点的好坏。或许我们可以根据统计学的方法，来获得最受大家喜爱的，好评度最高的景点，但要做到这一点，需要付出高昂的代价：事关景点好坏的因素非常多，而且比较零散且不集中，这意味着我们需要从很多不同的渠道来获得信息，而且需要很高的技术水平来处理及管理好这些数据。我们认为，这已经超出了我们的水平范围，我们是做不到的。【……………】

为景点安排行程的算法

所谓安排行程，就是指安排各个景点游览的时间以及先后顺序。在我们的设想中，在安排行程的时候，需要先按照地理位置对各个景点进行聚类，然后在安排顺序的时候，将聚合为同一类的景点前后连续地安排在一起。我们认为，通过这样的安排，能够减少游客花费在路途上的时间，给予游客更优的旅游体验。在类与类之间，以及同一类的景点之间进行时间先后安排时，则要借助上一步的排序结果：对类而言，使用的是类中各个景点排名的平均值；对一个类中的各个景点而言，则可以通过对比它们的排名来进行时间安排。

此处还有一点需要注意的，就是为每个景点所安排的时间：有的景点非常值得观赏，或者其本身就需要比较长的时间，那么我们就需要给这样的景点安排一天的行程；假如有的景点并不是这么重要，或者其本身并不需要太多时间，那么我们就只给这样的行程安排半天的行程就可以了。这无疑增加了“行程安排”这部分的复杂度。按照我们的设想，为了降低难度，我们的初步方案如下：只把一天的时间分成上午和下午两部分，然后为每个景点安排一个部分的时间，或者将一整天的时间都安排给一个景点。很明显，这个方案非常的原始，很有改进的空间，但是由于时间精力及能力有限，我们选择了这个比较简单的方案。

交通路线规划及食宿安排

当每个景点的行程安排好了以后，接下来这部分的工作就比较简单了。

对于安排交通路线来说，基本上一些地图应用就已经帮我们做好了这些事情，提供了丰富而大量的信息。我们需要做的，就只是在云端上将这些路线信息从那些地图网站上获取下来就可以了。

对于食宿安排这部分而言，似乎就要比上一点要复杂了。我们注意到，对于游客而言，他们行程的主线是往返于各个景点之间。因此，我们在为游客们安排食宿的时候，一定是在其游历景点的周边，结合其游历各个景点的时间进行安排的。正如上文提到的那样，在安排三餐的时候，我们需要关注的只是游客在用餐时间前后游览的景点，并且在那周围寻找餐厅就可以了。寻找餐厅的方法很简单，地图类应用已经帮我们做好了这个工作，只需要在景点附近搜索即可。搜索出来的餐厅也许会有很多间，我们还需要在所有搜索出来的餐厅里面挑选出最适合的。至于挑选的方式，根据我们初步的设想，是借助点评类网站来完成的：点评类网站上有大量餐厅的点评，我们会参考这些信息，为用户推荐景点附近的，而且综合评价最好的餐厅。

手机客户端的设计和制作

云端上要做的的工作，已经通过上面几点描述完毕了。最后还剩下一点，就是手机客户端界面的实现，以及客户端与云端之间如何进行通讯的问题。手机界面的实现，需要参照我们设定的使用流程。由于我们也是刚刚开始学习如何制作Android手机的应用，因此我们可能无法做出一个拥有华丽界面或是动画效果复杂的手机应用，最多只能保证其功能上的正确。至于客户端与云端之间的通讯，这应该不是一个大的问题，只要云端服务器有一个固定的IP地址，然后手机客户端在有网络的情况下与这个IP地址之间进行通讯即可。

2.4 开发计划及进度安排

【.....】