题目名称：安卓平台天津理工大学学生信息查询系统

手机应用名称：移动校园

Page 1

大家好，现在我来介绍下我的毕业设计。

这个系统的名称是“安卓平台天津理工大学学生信息查询系统”，手机应用名为“移动校园”。

Page 2

接下来我会简单介绍下这个系统的应用背景与意义、它给用户提供的服务、它的特色与优势、还有它的系统结构。

Page 3

首先来看看这个系统的立意或者说出发点。

Page 4

如果大家尝试用手机访问我们学校的网站的话，应该会感觉到其布局与交互过程对手机用户来说非常不便。就像这张图片里的效果，不仅难以查看，而且难以选择功能或点击按钮。网站页面比较复杂，用手机打开的话，加载和交互过程都很慢，也费流量。而且为了找到所需信息，往往需要跳转多个页面，不便性更是成倍增长。

Page 5

为在不修改影响现有系统的前提下改善这一情况，本系统应运而生。

本系统能够解析我校各网站上的各类信息，可以把解析到的信息缓存在专门的代理服务器上，最终可以通过专门的手机应用，快捷高效地为用户提供所需信息。

Page 6

事不宜迟，先来看看这个系统为用户提供的具体功能。

Page 7

用户可以用这个手机应用查询自己的学籍信息、入学信息等个人信息。

Page 8

也可以查看自己的课程表、成绩单和课程详情。

Page 9

还可以使用它查看各网站的各类通告事项。例如教务处的“重要通知”、“教务快讯”、“毕业设计相关规定”，计算机学院的“学生通知”、“学院新闻”，计算机学院学生网站的“通知公告”、“招聘快讯”、“下载中心”等等。用户可以根据标题、通知类别、发布者搜索通知。

Page 10

用户可以在设置页面设置查询账号、通知更新频率等配置。

Page 11

下边再介绍下本系统的一些特点和优势。

Page 12

本系统包含三个子项目。

网站内容解析器负责读取各网站的网页并解析出其中的信息。

解析代理服务器定时解析并保存各网站上的通用的非个人的公开信息。供代理用户使用。

手机客户端负责最终的信息呈现和用户交互。客户端会自动保存解析到的信息，可以离线使用。

手机客户端通过代理服务器，可以快速高效地得到通知列表，以减少用户等待时间，尽可能减少流量消耗。

解析器只读取必要的HTML网页，不访问JavaScript脚本、图片、Flash等文件，代理服务器只读取经过更新的通知列表，客户端通过代理服务器得到通知列表，避免了所有用户都扫描学校网站。因此本系统也能帮学校网站减轻负担。

Page 13

本系统的代理服务器使用Servlet框架实现，目前支持GAE应用服务器。这是它在GAE上的运行效果，这是仪表盘、日志、数据存储。

Page 14

这里总结下本系统的优势。

一方面，手机应用仅读取必要的信息，不下载多余文件；另一方面，对于较难解析的通知列表，手机客户端从服务器读取解析后的“干货”。网络操作少并且集中，更新快，流量和电量消耗较少。

使用此应用，用户只需一两步简单操作即可进入所需页面。应用自动显示当天的课程安排，可以查看自己的学籍等个人信息，可以使用查找功能查询各网站各类别的信息。

用户可以随时随地快捷便利地查询所需信息。

此外，应用还在本地保存抓取到的信息，可以自动更新通知，减少了重复的网络操作。用户可以修改本地数据，添加附注，实用性得到提高。

Page 15

下边来看看系统的结构。

Page 16

这个系统的目标用户是我校的学生，正如之前提到的，用户可以使用本应用查看个人信息、课表、成绩和通知。

Page 17

①本系统有三个组件：网站内容解析器、代理服务器和手机前端。

②解析器同时部署在代理服务器和手机上，代理服务程序部署在Servlet应用服务器上，手机前端部署在手机上。手机前端使用解析器从学校网站解析信息，手机前端使用Hessian ['hesiən]协议通过互联网从代理服务器得到通知的解析结果。

③这三个组件的依赖关系如图所示。代理服务器和手机前端都依赖于解析器，手机前端依赖于代理服务器。底下三个是第三方库，Jsoup用于读取网页并把其转换为DOM文档对象。代理服务程序依赖于Hessian协议库。Hessian是一种二进制远程过程调用（RPC）协议。

Page 18

先来看看解析器的设计。

解析器项目中最重要的是ConnectionAgent连接代理和Parser解析器类，这是它们的类图，幻灯片上的字有点小，可以看说明书附录1中的大图。

ConnectionAgent使用了代理模式，它代理了Jsoup中的Connection类的get、post等操作。在其基础上各ConnectionAgent具体类封装了各网站特有的访问方式，包括登录网站的过程。

Parser解析器使用ConnectionAgent得到所需页面的DOM文档对象后，对其作进一步解析，最终得到Course课程、Post通知等实体类。

根据解析内容的不同，Parser衍生出个人信息解析器、课程解析器和通知解析器，通知解析器根据访问的网站再进一步特化，课程解析器又衍生为已选课程解析器和成绩解析器。各具体解析器，根据其解析的内容，依赖于对应网站的ConnectionAgent类。

Page 19

这里以解析已选课程信息为例，看看这些类的对象的交互过程。

解析器用户实例化已选课程解析器和师生服务网站连接代理后，告诉连接代理查询账号ID和密码，再把设置好的连接代理对象传给解析器，即可让解析器开始解析后。解析器首先用连接代理读取已选课程页面，如果连接代理发现还没有登录，它会先自动登录网站，然后读取并返回指定网页，最后解析器再从网页抽取出已选课程信息，构造出相应的Course实体对象，返回给用户。

Page 20

使用时的具体代码类似这样。

可以先准备好要用到连接代理对象并设置好必要信息，如账号名和密码。

然后根据所需信息创建对应的解析器对象，设置连接代理后，即可进行解析操作。

Page 21

例子的解析结果类似这样

Page 22

通知解析器可以设置一些过滤器，例如这个例子设置了最早通知时间，只解析指定时间以后的通知。

Page 23

代理服务器包括服务API、服务器端和客户端。API作为公约，确定了服务器为客户提供的服务和使用方式。服务器端包括后端更新器和前端接口，前者自动更新通知，后者为客户提供服务接口。客户端为用户提供了必要的工具库。

Page 24

最后看看手机客户端的导航层次设计。

用户可以在主界面选择所需功能，然后跳转入相应界面。在课程表和成绩单界面中点击任意课程，可以进入这个课程的详细信息页面。在通知列表界面点击任意一条通知，可以查看这个通知的详细内容。第一次进入通知详情界面，会自动下载相应通知的正文并在本地保存，以后不必重复下载。