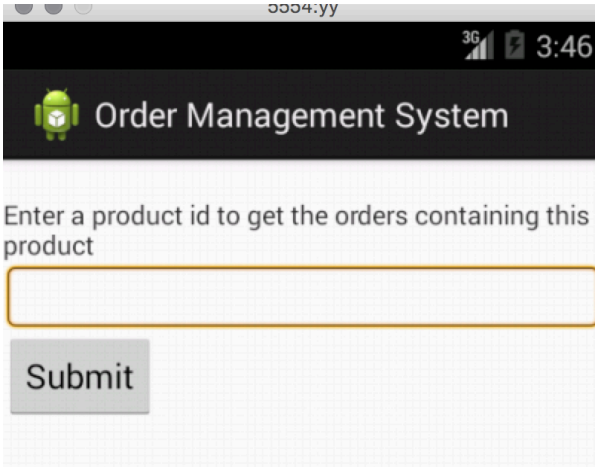
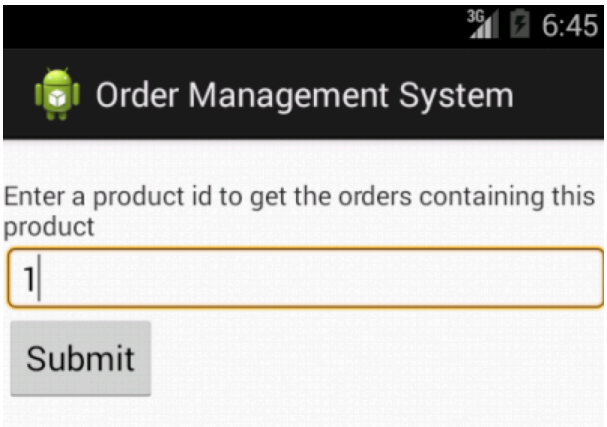
Project 4 Task 2 – Order Management System
By Tianyue Xiao tianyuex@andrew.cmu.edu

Description:
My application takes a search product id from the user, and uses it to fetch and display orders containing that product from OData.

Here is how my application meets the task requirements
1. Implement a native Android application
The name of my native Android application project in Eclipse is:
Project4Android
1.1. Has at least two different kinds of views in your Layout (TextView, EditText, ImageView, etc.)
My application uses TextView, EditText and Button. See main.xml for details of how they are incorporated into the LinearLayout.
Here is a screenshot of the layout before the orders has been fetched



1.2. Requires input from the user
Here is a screenshot of the user searching for orders of product 1



1.3. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does an HTTP GET request in GetOrder.java. The HTTP request is:
http://1-dot-tianyue7777777.appspot.com/project4?id= + searchTerm
where searchTerm is the user's search term.
The search method makes this request of my web application, parses the returned Json to find the orders and display them

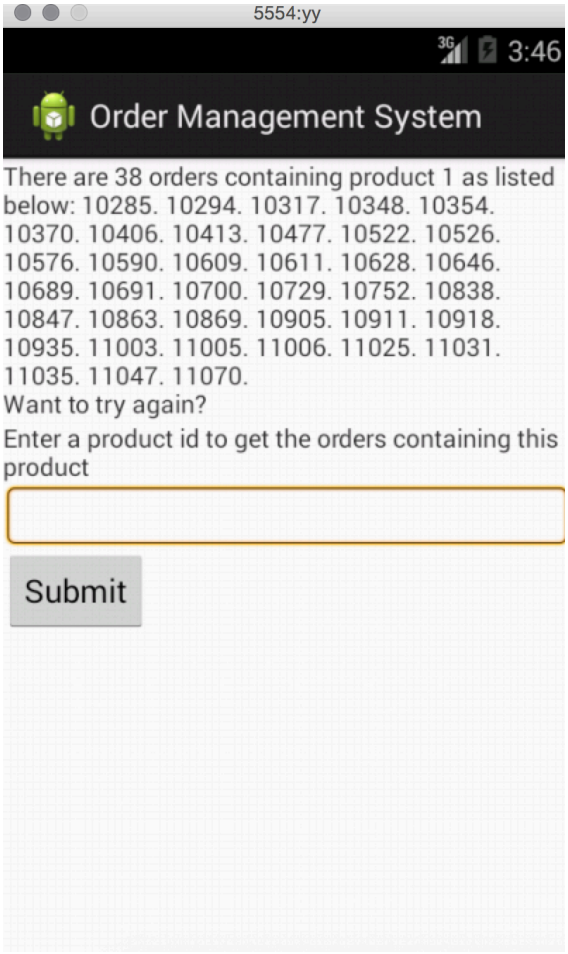
1.4. Receives and parses an XML or JSON formatted reply from the web service

An example of the Json reply is:

{"result":"There are 38 orders containing product 1 as listed below: 10285. 10294. 10317. 10348. 10354. 10370. 10406. 10413. 10477. 10522. 10526. 10576. 10590. 10609. 10611. 10628. 10646. 10689. 10691. 10700. 10729. 10752. 10838. 10847. 10863. 10869. 10905. 10911. 10918. 10935. 11003. 11005. 11006. 11025. 11031. 11035. 11047. 11070. "}

1.5. Displays new information to the user

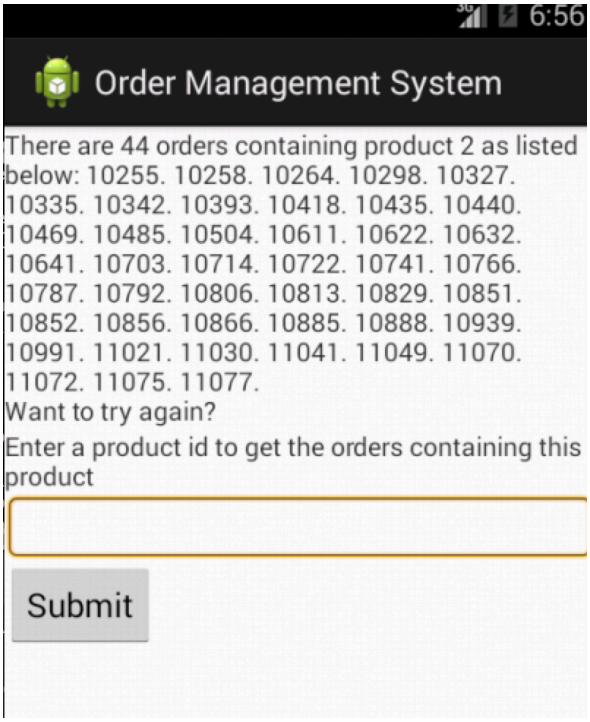
Here is the screen shot after the orders have been returned.



1.6. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another product id and hit Submit. Here is an example of

having typed in "2".



2. Implement a web application, deployed to Google App Engine

The name of the Google App Engine project in Eclipse is:
Project4

2.1. Using an HttpServlet to implement a simple (can be a single path) API

The controller is Project4Servlet.java
The model is Project4Model.java

2.2. Receives an HTTP request from the native Android application

Project4Servlet.java receives the HTTP GET request with the argument "id".

2.3. Executes business logic appropriate to your application

Project4Servlet.java makes an HTTP request to:
"http://services.odata.org/Northwind/Northwind.svc/Products("
+ id + ")/Order_Details?$format=json"
It then parses the Json response and extracts the parts it needs to respond to the Android application.

2.4. Replies to the Android application with an XML or JSON formatted response.

Project4Servlet.jsp formats the response to the mobile application in a simple Json format of my own design:
{"status": "success", "result":"There are 38 orders containing product 1 as listed below: 10285. 10294. 10317. 10348. 10354. 10370. 10406. 10413. 10477. 10522. 10526. 10576. 10590. 10609. 10611. 10628. 10646. 10689. 10691. 10700. 10729. 10752. 10838. 10847. 10863. 10869. 10905. 10911. 10918. 10935. 11003. 11005. 11006. 11025. 11031. 11035. 11047. 11070. "}