Course Project Description and Overview

CIS 550 — Fall 2007

September 4, 2007

1 Overview

In this course, we will have covered not only how databases are useful in managing structured information, but also how they can be used to facilitate information retrieval (search), and how database techniques can be useful in *integrating* data.

In the commercial/open-source space of Web sites and Web applications, special-purpose information integration is often performed using *mash-ups*. (The term derives from music mash-ups, which sample from others' recordings.)

A Web mash-up is typically built over XML-based Web services provided by sites such as Google, Amazon, etc., and it assembles content in an interesting application-specific way. A humorous example of this is the New York City Smell Map, http://www.gawker.com/maps/smell/.

For the course project, you will use the skills you learned related to XML and XQuery, relational database design, SQL, and Java programming in order to provide an apartment search mashup for Philadelphia. This mashup will integrate maps from Google Maps with apartment rental listings from Craigslist and the Philadelphia Inquirer. (For extra credit, you will also integrate Flickr photos in the proximity of the listings.)

2 Requirements

Your mashup service will broadly include the following modules:

- **User login, authentication, and management**. You should be able to support both *unregistered* users, who do not log into the system, and *registered* users. They should also have the ability to change their password, add or remove bookmarks, and so on.

- **Apartment-search screen**. Users will have the ability to query based on a combination of keywords and attribute constraints (price, location, number of bedrooms, pets allowed). Keyword searches will be expanded to include synonyms. The results will be projected onto a Google Map and should also have clickable "detailed" entries.

- **Apartment detail view**. When a user clicks on a listing, he or she should be taken to a detailed view that shows the apartment listing, a street-level map of the apartment, and a list of other query results that share the same zipcode. For extra credit, photos from Flickr that are geocoded to be within the same neighborhood should be shown.

- **Bookmarks**. Queries can be saved as "bookmarks." For unregistered users, bookmarks will expire with the session. With registered users, when they log in, they should see a summary page of all listings related to their bookmarks (updated to the current time). Users should be able to remove bookmarks or edit the queries that lie underneath them.

- **Listing fetcher / crawler**. You will need to build a tool to go to www.phillyforrent.com and extract listings from the HTML in order to populate a database of rental listings. Likewise, you will need to build a tool to fetch the philadelphia.craigslist.org/apa/index.rss RSS feed for apartment listings, and to try to extract the location, price, etc.

- **Data warehouse / index**. All listings will be archived in a database, with entries for price, number of bedrooms, location, contact info, and so on. They will additionally be indexed by keyword so users can search. We expect the keyword inverted index to be stored in relational tables.

3 Example

A simple example that should give you some ideas is:
http://www.housingmaps.com/
However, your project will go beyond this in terms of capabilities, including user accounts and saved bookmarks.

4 Implementation

We expect you to use Java Servlets running on the SEAS Apache Tomcat server for your project. We also expect your listing fetcher to convert from HTML to XML before loading the data into the data warehouse (which should be a database on the SEAS Oracle instance used for your homework assignments).

We expect that the keyword search capabilities will be implemented using an inverted index implemented over relational tables, and that your keyword search will be based on SQL queries.

Additionally, you may use Eclipse and SAXON (installed in the computer labs) to do your Java and XQuery development.