CSE 484 Project Proposal

1) Team:
   Aaron Nash  
   anash@uw.edu  
   GitHub: aaronnash
   
   Josiah Adams  
   josiah@josiahadams.com  
   GitHub: JosiahAdams

2) Problem:
   Mobile devices provide a plethora of sensors and contextual data that is otherwise not available in most computer systems. Currently, RevMiner's filtering is only done through keyword parsing and requires a well-crafted query to produce valuable results. It is our goal to expedite the process of finding dining locations while providing high quality results. To do this we plan to make RevMiner location aware and provide an interface that is both intuitive and detailed. At the completion of the quarter we will have a product that allows users to quickly find nearby dining locations with the ability to refine searches through locational constraints and restaurant features (the latter being currently available through RevMiner).

3.) Artifacts:
   - The Android phone app package
   - a web-only (HTML5) user interface and its source code (modification of existing desktop web interface)
   - Source code
   - End-user documentation
   - Developer documentation (i.e. API documentation)
   - GPS Coordinates for businesses (and code used to obtain)
   - Android GUI Mockups

4.) Artifact Creation
   - We’re going to stick with node.js for the webserver and mongodb for the database. The native Android app will be written in Java. The code to get GPS coordinates will be written in Python.
   - Client server communication for the Android app will run over HTTP encoded in JSON.
   - Initially we’re going to host our artifacts locally. We’re looking into use of Joyent or Heroku for later iterations.
   - Data for GPS coordinates will be gathered through Google’s APIs.
   - There will be a significant effort given to developing a mobile optimized version of the GUI. We will develop mockups for this by the first milestone.
   - For version control we’ll both use GitHub.

5.) Division of Labor & Milestones

1. Date: January 26, 2012
a. Complete backend
   i. Modify database schema to contain GPS coordinates of each restaurant. [Josiah]
   ii. Convert address for each restaurant into GPS coordinates. [Josiah]
   iii. Client-server communication protocol specification finalized for v1 [Josiah]
   iv. Query engine rewritten to handle locational constraints [Aaron]
   v. Server query processor and data provider implemented and tested. [Aaron]
i. Android GUI mock finished [Aaron]

2. Date: February 16, 2012
   a. Document internal interfaces that will tie together MVC for Android app [Josiah &
      Aaron]
b. Android client capable of requesting and processing data from server. [Josiah]
c. Android GUI model and view implemented (controller may be incomplete) [Josiah]
d. Android client capable of interacting with Google Maps API. [Aaron]
e. Complete documentation [Josiah & Aaron]

3. Date: March 6, 2012
   a. Android app fully integrated and completed. [Josiah & Aaron]
b. Complete HTML implementation. [Josiah]
c. Integration tests written for Android app. [Aaron]