CS 190 Project Requirements Proposal
Amy Simpson (asimpson)
Feb. 10, 2006
GeoEvent

1. **Title: GeoEvent**

2. **Description**
   GeoEvent is designed for Rhode Islanders looking for something to do. Often people want to go out and do something but lack the motivation, time, or energy to put in the effort to actually find that something. Newspaper listings lack the ability to adapt to any of the specific needs of the user and can be hard to use. GeoEvent will be able to mesh the current events listings from the newspaper (the Providence Journal since it is Rhode Island based) with the Google Maps API to provide a much more interactive and intuitive way to look for interesting activities. One current example of a similar project can be found at http://api.local.yahoo.com/eb/demo/. GeoEvent will go beyond, however, and provide additional useful features (specified in section 3 below).

3. **Feature Set/Priorities** *(Rated 1-6: 1 being the highest priority)*
   (The priority rating is a result of responses to questions from possible users and an attempt to mirror similar existing interfaces in order to increase the intuitiveness of the project's use)
   a. Google Maps API with flags marking the location of local events that qualify for the filter parameters provided. [1]
      i. This will be a frame on the page containing a Google Map of the area. When GeoEvent is first opened up, the default map should be of Providence and all filter types of events should be included on the map. All events are marked by a flags which can be clicked for details (more details in section c below.)
   b. Be able to enter in a 'base' address (the address the user will be starting from, for example their home address or the address of their office if they are leaving from work [2]
   c. Clicking on an event:
      i. an event's flag should expand to show the brief details of the event in a concise, readable, and consistent manner.
         1. Display the event location [2]
         2. Display the event time [2]
         3. Display the price of the event [2]
         4. Show the distance from specified base address (if there is one, other wise this field should not be included in the flag information) [3]

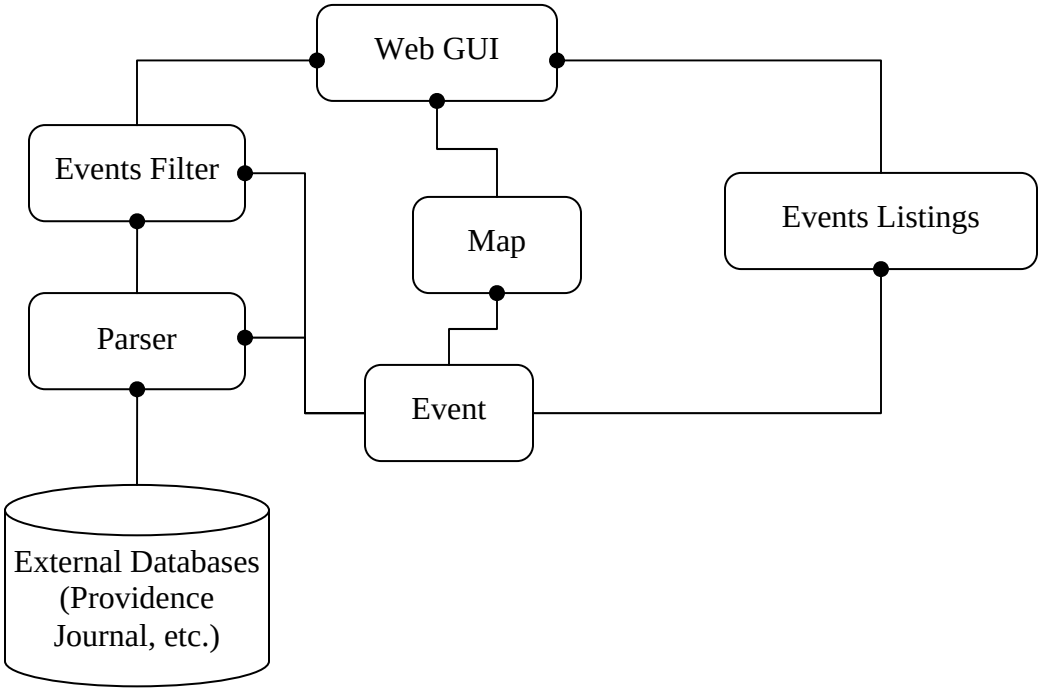
d. An interface (preferably, and most intuitively a calendar) for choosing the dates to search over[2]
e. Ability to filter types of events (e.g. choose only Theatre & Museum) [2]
f. Be able to get directions to a specific event [3]
g. Display below the map window a list of the filtered events [1]

Optional Features (Priority four and over)

(a) Have an interface that will learn from past usage of specific users and be able to 'suggest' events [6].
    (i) Using past inputs (searches and events the directions were found for) from the user, GeoEvent would be able to find and 'suggest' similar types of events that the user might enjoy
(b) Be able to select more than one date at a time [4]

The first priority (priority one) is having a map that shows and lists the events around Rhode Island. This is absolutely essential. For this project to have real usefulness, all items in priority two should be implemented. This includes event details including time, location, and price. The third set of priorities are extremely useful and what would make this project more useful than what is currently available since it would integrate the direction creating capabilities of Google Maps into the project, but is not essential to the overall functionality of the project. Beyond that, other items are helpful and added bonuses, but not what truly define the project.

4. System Model Diagram



This is the graphical user interface on the web that the user will interact with *(see GUI diagram and description below for more details)*. It will contain an input box for the window, a map interface (it will use the Google Maps API), filtering input, and a list of events.

- filterEvents(selectedEvents, date)
  - This will filter the events to be displayed based on the selections of the user. It will be performed in response to a user clicking filter.
- getDirections(event)
  - This will obtain directions to a specified event
- getDistance(event)
  - Given an event, this should calculate the distance of a specific event to the base address
- refocus(address)
  - Given an address, the map GUI will recenter the map to that address

Events Filter

This obtains events from the parser and filters them.

- getEvents(selectedEventTypes,date)
    - returns a list of all events that are found matching the filter selections.

Parser

The Parser parses external databases including the Providence Journal for the events. Since these events do not change often, the Events Filter does not need to obtain them often – all events are returned.

- getAllEvents(date)
    - Returns a list of all the events in the database for a specific date

Map

The Map handles all interaction with the Map API

- setEvents(events)
    - This sets the events that should be available on the map
- refocus(latitude, longitude)
    - This recenters the map to the specified location
- The map responds to panning, zooming, and clicking on specific events.

Events Listings

This displays within the GUI the list of events.

- setEvents(events)

Event

An Event holds all the data for every event. It contains all the relavent details.

- getTime( ) – returns the time of the event
- getDate( ) – returns the date of the event
- getLocation( ) – returns the location of the event
- getPrice( ) – returns the price of the event

5. GUI Diagram

The GUI is organized so that the most noticeable item is the map itself. Easily accessible on the right are the options for changing and filtering the events. They are in a standard clean format so that they do not distract from the map. Below this main part of the GUI, the list of events is displayed.

The items in the map, corresponding to the green number tag, are:

1. This is the map that will display the event locations. Clicking on a tag will provide the user with more complete information about the event including time, location, cost, and distance (see details in the features section above)
2. This area is where the user inputs the address to center around and base directions from.
3. This box contains all the filter options for event types.
4. This calendar allows the user to choose a date to look at events for.
5. All events and their corresponding details are displayed here.

6. **Usage Requirements**
    a. Since this product would be of public interest it would need to be reliable under a high usage (1000+).
    b. Any longer of a response time than 5-10 seconds would be inadequate since multiple searches or refinements of searches (different types of events, the zoom factor on the map) could be made in short succession.
    c. GeoEvent must be consistent.
        i. If a user searches around a base address and certain type filters and finds an event they like, if they come back an hour later to get directions, they should easily be able to find it with the same search parameters.
        ii. Essentially, searches with the same parameters should return the same results.
7. **Non-Functional Requirements**
    a. Testing
        i. Besides testing to ensure that the project responds in the specified amount of time, it is essential that it continues to do so as the user usage increases.
        ii. Data displayed must match events included in the database
            1. This should be tested under a high load for reliability
    b. GeoEvent must be accessible from any computer through the web
    c. Must be written in C++
    d. Must be written and ready to demo by the end of April 2006
    e. Must have thorough documentation on the use of the program and possible extensions for the future that were not implemented in this version
    f. Ease of Use
        i. This project should be usable for anyone capable of surfing the internet.
8. **Divisibility**
    a. This project should end up being sufficiently divisible for the projected size group for this year's class. Different elements include
        i. Overall GUI

ii. Google Maps API interaction
iii. Parsing of the Providence Journal's Events Page information and other events listings
iv. Filtering and searching of the event information once parsed

9. Specific Challenges/Issues for the Project

a. External Dependencies
   i. It is unclear at the moment how well organized the Providence Journal's Events information is (it is not expected to be highly so). The largest difficulty could lie in deciphering this language so that it is in a useful enough of a format to use.
   ii. **Neither the Google Maps API nor the Yahoo Maps API interfaces with their directions creating capability.**
   iii. Google forces the use of an external geocoder (switches an address to longitude/latitude that can be used in the API). Yahoo has an internal geocoder.