# Specification Document: GeoEvents
By ychatura

## Project Description

### Overview
- GeoEvents is a web-based program that allows users to search for local events and visualized them spatially. With intuitive and easy-to-use interface, users can narrow down the search by events filters such as dates, types of events, and distance. GeoEvents will parse the Providence Journal's event listing page, get the map from the Google Maps API and then display it.
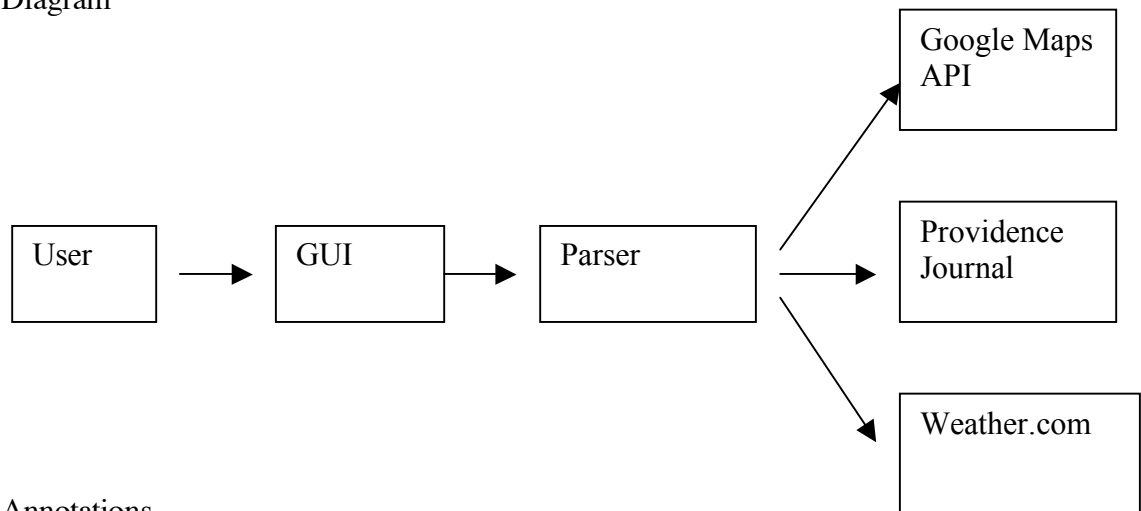
### Target Users
- Although GeoEvents will be particularly appealing to Rhode Islanders, the target users are the general public. GeoEvents will save them time and effort to get the details from various sources by putting them together in one place.

### Updated Features with Priorities (1 is the highest priority)
- Google Maps API
  a) When the users first open up GeoEvents, the map of Providence will be displayed as default. [1]
  b) After the search, the map will display each event as bubbles on the locations at those events. Bubbles will contain brief information about the events. [1]
  c) If a user's address is entered, it will also be displayed on the map with driving directions. [3]
  d) When an event is clicked, complete detail of that event will be displayed in a separate window. [2]

- Events Filters
  a) Calendar: allow users to choose a particular day or range of days [2]
  b) Distance: allow users to enter maximum distance from home address to events in miles. [3]
  c) Types of events: allow users to select as many as they want from category list. [2]

- Event list and driving direction window
  a) Display the result of the search with specified filters. [4]

- Weather forecast
  a) Display the weather forecast with temperatures from weather.com for the specified days. [4]

- Personal Events
  a) Allow users to post their personal events so other users can find it. The user must provide information about the event such as dates, time, location and costs. [5]

- Suggestions
  a) Require every user to sign in with their account and keep the history of their past searches. Based on search history, make a suggestion on the event. [6]
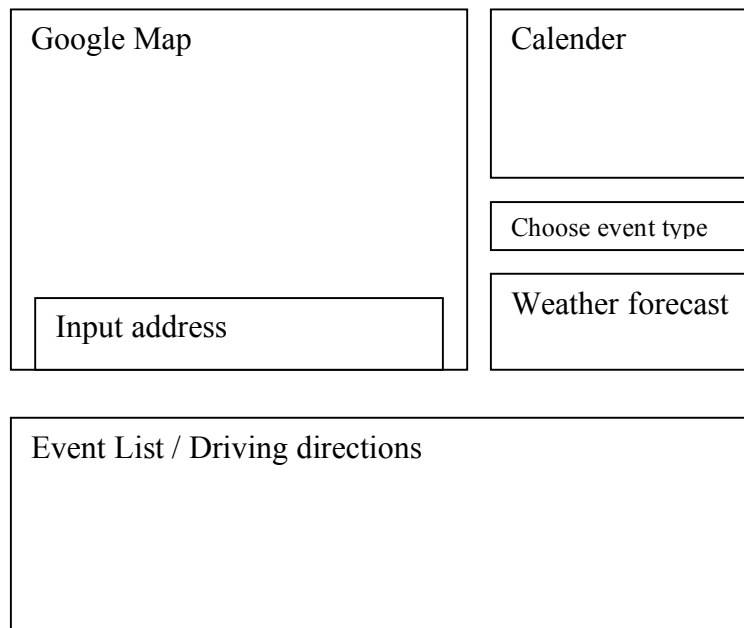
System Model

Diagram

```
┌─────────┐      ┌─────────┐      ┌─────────┐              ┌──────────────┐
│  User   │ ───► │  GUI    │ ───► │ Parser  │ ───────────► │ Google Maps  │
│         │      │         │      │         │              │ API          │
└─────────┘      └─────────┘      └─────────┘       │      └──────────────┘
                                                    │
                                                    │      ┌──────────────┐
                                                    ├────► │ Providence   │
                                                    │      │ Journal      │
                                                    │      └──────────────┘
                                                    │
                                                    │      ┌──────────────┐
                                                    └────► │ Weather.com  │
                                                           │              │
                                                           └──────────────┘
```

Annotations
- GUI
  - GUI will take the information about the filters that users enter and pass them to the parser.
- Parser
  - After receiving search criteria from GUI, parser will go to Providence Journal, parse the event page, and get the detaoils of the events
  - From the location of events it has found on the Providence Journal, it will get the map from Google Maps
  - Get the weather forecast of specified days from weather.com
  - Return the results from parsing the Providence Journal that match search criteria to GUI with Google map

User Interface Diagrams

```
┌──────────────────────────────┐   ┌──────────────────────┐
│ Google Map                   │   │ Calender             │
│                              │   │                      │
│                              │   │                      │
│                              │   │                      │
│                              │   └──────────────────────┘
│                              │   ┌──────────────────────┐
│   ┌──────────────────────┐   │   │ Choose event type    │
│   │ Input address        │   │   └──────────────────────┘
│   │                      │   │   ┌──────────────────────┐
└───┴──────────────────────┴───┘   │ Weather forecast     │
                                   │                      │
                                   └──────────────────────┘

┌────────────────────────────────────────────┐
│ Event List / Driving directions            │
│                                             │
│                                             │
│                                             │
│                                             │
└────────────────────────────────────────────┘
```

Non-Functional Requirements

Performance
- The waiting time after the search should not exceed 15 seconds.

Testing
- Must test if the map is displayed correctly.
- The program must be consistent: searches with the same inputs should return the same results
- Must not display events that do not exist.

Reliability
- The program should not crash if an error occurs.
- 500 users should be able to use the program at the same time.

Ease of Use
- The interface must be intuitive and simple enough for computer novice to use

Portability
- GeoEvents is a web-based program. Anyone with a web browser and Internet connection can use the program.

Documentation
- The documentation should be updated throughout the project to allow group members as well as the clients to know about the progress.
- The documentation should be well-written, concise and easy to understand.

Dependencies on other systems
- Google Maps API
- The providence Journal website


Risky Parts
The risky part is to come up with intuitive layout of GUI and to deal with other systems such as the Providence Journal website