Local Events Application

Team Members: Jennifer Soh || Brian Chin

Introduction

* 1. **Foreword**

This application provides a single resource for discovering local events that people might be interested in. This application is meant to serve people who are looking for something to do or just want to try something new. This service would be similar t to Meetup in that people can find local events to participate in and try something new. The differences between our application and Meetup are that a membership is not required, there are no groups that you would have to join or request to join, and you can produce a list of local events based on interests or just location and participate in any of them.

This application would work accepting a users location input and produce a list of events within that area that have been posted on Eventful API.

Building the application involves coding the application in jQuery, creating a user interface with jQuery Mobile, connecting the application to the Eventful API, then packaging it using PhoneGap.

# 2. Project Organization

## 2.1 Development Model

Home Page:

Introduction

Go to Search Page > Search by Location and Keyword

Store Events > See Saved Events

Search Page:

Enter your location.

Enter a keyword.

Saved Events:

List of Saved Events > View Event Details

Manage Events > Delete Events from Saved Events

## 2.2 Project Team

## 

The table below lists the members of the EVENTS project team. The project Team consists of 2 members

|  |
| --- |
| **Project Team** |
| User Interface: Brian / Jennifer |
| Architecture: Brian / Jennifer |
| Backbone.js: Jennifer |
| Mobile Devices Tester: Brian |
| Desktop tester: Jennifer |
| jQuery Prototype: Brian |
| PhoneGap: Brian |

# 3. Risk Analysis

### 3.1 Risk Analysis

|  |  |  |
| --- | --- | --- |
| **Risk** | **Affects** | **Descriptions** |
| APIs | Entire app is useless without the information | API: Eventful  The API is needed to find and return event information.  Inexperience with APIs and cannot connect them to our app |
| Backbone.JS | Communication with the API | Without Backbone.js, pages won’t load dynamically and require refresh. |
| Time | Incomplete Project | Our concern is that we won’t be able to fully implement all the features our application would need to be successful |
| jQuery | Won’t be able to code anything functional | jQuery is required to create any and all functionality for our application in Backbone beyond the jQuery plugins we will use to build the interface. |
|  |  |  |

### 3.2 Risk Analysis (i)

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Effects** |
| **APIs** | 80% | The entire app is not functioning properly and will be unable to get information. |
| **Backbone.js** | 75% | The app will not function properly |
| **jQuery** | 50% | The consist a good proportion of the app, and without it the app will not work |
| **Time** | 90% | If we don’t finish in time most of application would be incomplete and have less features |
|  |  |  |

### 

### 3.2 Risk Analysis Strategy

|  |  |
| --- | --- |
| **Risk** | **Solution / Mitigation** |
| **APIs** | Look at tutorials, ask for assistances, read books about APIs |
| **Backbone.js** | Understand the concept, practice coding with backbone, ask for help if required |
| **jQuery** | Practice jQuery, look at other examples and follow them. |
| **Time** | Use time span wisely, focus on goals , set dates for things to be done. |
| **PhoneGap** | If PhoneGap doesn’t work, our application can’t be converted to a Mobile Application. |

# **4. Hardware and Software Requirements**

Hardware:

Build code on: Laptop, Desktop computers

Run code on: Mobile devices, desktops and laptops.

Software:

Build could on: coda2, komodo

Run code on: Safari, Google Chrome

# **5. Schedule and Work Breakdown Structure**

Week 1: Clearly assess roles, capabilities, continue research and begin code structure.

Week 2: Start on building the site jQuery Mobile, begin basic core structure of higher functions in Backbone.

Week 3: Dealing with API

1. Begin with the prototype to figure out the API and structure.
2. Research the API to understand what it can provide.
3. Create other filtering functions

Week 4: Clean up and test application, improve and, if possible, further develop the application.

Week 5: Last week for testing