Software Requirements Specification

For

DragonEvents

|  |  |
| --- | --- |
| **Instructor:** | David Augenblick |
| **Team Members:** | Saffat Hasan, Harley Sacramento, Haley Lee, Nicholas Nguyen, Rahul Godaba |
| **Cycle:** | 01 |
| **Date Submitted:** |  |

Document template copyright 2005-2015, CCI Faculty. Version 2.3. Use permitted under Creative Commons license CC-BY-NC-SA. See http://creativecommons.org/licenses/by-nc-sa/3.0/.

Grading Rubric - Requirements Specification

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading. Change is possible, especially given the dynamic nature of this course. Any change will be applied consistently for the entire class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Achievement** | **Minimal** | **Exemplary** | **Pts** | **Score** |
| **Content (80)** | Section(s) missing, not useful, inconsistent, or wrong. | Provides all relevant information correctly and with appropriate detail |  |  |
| Introduction  Scope  Definitions |  |  | 10 |  |
| User Profile |  |  | 20 |  |
| Functional Requirements |  |  | 30 |  |
| Performance & Design Requirements |  |  | 10 |  |
| Data Requirements |  |  | 10 |  |
| **Writing (20)** |  |  |  |  |
| Grammar and Spelling | Many serious mistakes in grammar or spelling | Grammar, punctuation, and spelling all correct | 10 |  |
| Expression | Hard to follow or poor word choices | Clear and concise. A pleasure to read | 5 |  |
| Tone | Tone not appropriate for technical writing | Tone is consistently professional |  |  |
| Organization | Information difficult to locate | All information is easy to find and important points stand out | 5 |  |
| Layout | Layout is inconsistent, visually distracting, or hinders use | Layout is attractive, consistent, and helps guide the reader |  |  |
| **Late Submission** |  |  | -10  -25 |  |
| **Total** |  |  | 100 |  |

# Introduction

This section provides an overview of DragonEvents. Scope provides a short description of the product and how it is useful; Definitions explains terms with which a reader may not be familiar; and User Profile identifies the ways different groups of people would make use of DragonEvents.

## Scope

DragonEvents is a website created for events that are taking place on or near the Drexel University campus. The events will allow users to be notified about any events and activities that they can participate in for no cost.

## Definitions, Acronyms, and Abbreviations

An event host is defined as the account that submits an event to DragonEvents.

## User Profile

-DragonEvents is targeted at Drexel students. While anyone can browse DragonEvents, Drexel students are allowed to submit events

Students will be able to submit any events and activities taking place on or near campus. They can set up their notification according to category or interests. The users can give feedback on each event.

- Premium accounts: non-Drexel students can submit events for a fee. This would, for example, allow local companies to submit advertisements for local, Drexel-targetted events.

# External Interfaces

This section identifies ways in which DragonEvents interacts with people and other systems.

## User Interface

DragonEvents will interact with users through three interfaces. The first interface will list events with options to both filter and sort events by several parameters (such as date, location, type). The second interface will allow users to submit events with accompanying features. The third interface is to allow users to manage or modify events that they have submitted.

## Data Interface

DragonEvents does not need data from external sources.

# Specific Requirements

## Functional Requirements

The statements below define the functional requirements for the system.

1. Sort and Filter events

DragonEvents needs to allow users to see events relevant events to them. For example, a user might browse DragonLinks in order to find events in the upcoming weekend (date range) with free food and music (event tags ‘free food’, and ‘music’).

1. Receive Event Submissions

DragonEvents needs to have a form to allow users to submit a new event. In addition, the system must also be able to filter malicious or false submissions to reduce spam.

1. Allow for event hosts to edit their own events.

Event hosts should be able to update their events (date, time, location, information)

1. Premium accounts / events

Although DragonEvents is targeted at Drexel students, non-Drexel students can submit events for a fee. The fee prevents spam from being generated and will be done through PayPal and automated. These events will be given distinction on the main page. These users are unable to select ‘interested’ or ‘going’.

1. Allow users to ‘interested’ or ‘going to’ an event

Users are able to select ‘Interested’ or ‘Going to’ (not both) for a given event. This is used as a metric for others to gauge how popular an event is.

1. Allow users to sign up to submit events

Drexel students are given the option to sign up in order to further interact with the website (submitting events or selecting ‘interested’/’going to’ for an event)

1. Allow users to report events

The DragonEvents relies on its users to ensure that all events are legitimate

## Performance Requirements

The statements below define the performance requirements for the system.

1. Response time

The response time is surely important with our website so we plan to have a maximum 3 second response time.

2. User Friendly

UI has to be user friendly, allowing for users to navigate the site to events tailored to them.

## Design Constraints

## Constraint: GPS

Reason: The GPS is a constraint because our project is a web-based meant for Drexel students so it is unlikely that we will implement it.

## Data Requirements

1. **Events**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| Location | String | no max | Where the event will be held |
| Event ID | Integer |  | Unique Identifier |
| Date | Date (without time zone) |  | Specific to Drexel so time zone is not needed |
| Tags | Integer Array | no max | references Tag table |
| UserID | Integer |  | references user account ID |
| is active | Boolean |  | Flag events as deleted |

1. **Tags**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| TagID | Integer |  | Unique Identifier |
| Description | String | No max | describes tag (e.g. music, food, free) |

1. **Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| UserID | Integer |  | Unique Identifier |
| First Name | String |  |  |
| Last Name | String |  |  |
| Email | Integer |  | must be stripped and lowercased and validated tg |
| is active | Boolean |  | users are required to be verified via email before being active |
| validation token | string | 32 | used to validate an email |

1. **Event information table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| EventID | Integer |  | references event table |
| UserID | Integer |  | references user table |
| Attending status | Enum |  | ‘Going’ ‘Interested’ or Null |

1. **Event report table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Size** | **Comment** |
| EventID | Integer |  |  |
| UserID | Integer |  |  |
| Comment | String |  | Allows user to describe report |