Crowd Science

Senior Design Final Documentation

Crowd Science Mapper

Hannah Aker Jiasong Yan

November 15, 2015

Contents

List of Figures

List of Tables

List of Algorithms

Overview Statements

0.1. Mission Statement

Mission statement inserted here. Does our team need a mission statement?

0.1.1. Product Description

The Crowd Science Mapper will be a generic crowdsourcing website, where ordinary citizens can report events such as butterfly or bird sightings, and view these events.

0.1.2. Goals

Distribute finalized toolkit or API to researchers and general public by 2022.

0.1.3. Assumptions

- Website
- Android App
- iOS App
- Queing mode for data transmission
- Lower power

0.2. Elevator Pitch

Crowd Science (also known as Citizen Science) aims at soliciting ideas, findings, and information to support scientific research, which could reduce time and cost for data acquisition and enhance the accuracy and generality of research results. This project was inspired by USGS Did You Feel It (DYFI) project, USGS Butterflies and Moths of North America (BAMONA) project, and Cornell's eBird project.

The Crowd Science Mapper will be a generic crowdsourcing website, where ordinary citizens can report events such as butterfly or bird sightings, and view these events. Administrators in this system will be able to approve individual events and modify event sets, with no programming experience using a graphical user interface.

Document Preparation and Updates

Current Version [1.2.0]

Prepared By:
Hannah Aker
Team Member #27

Revision History

Date	***Author***	***Version***	***Comments***
9/14/15	*Hannah Aker*	*1.0.0*	*Duplicated design template for editing*
9/30/15	*Hannah Aker*	*1.1.0*	*First draft of requirements section*
10/30/15	*Hannah Aker*	*1.2.0*	*Add sprint reports, refine system template, refine requirements, filled in project and mission/elevator section*
11/14/15	*Hannah Aker*	*1.2.0*	*Finished requirements.*

1. Overview and concept of operations

1.1. Scope

This section covers an overview of the Crowd Science Mapper. This section will contain information about the prupose, goals, major system components, and technology used.

1.2. Purpose

The purpose of of this system is to provide a generic crowdsourcing website, where ordinary citizens can report events such as butterfly or bird sightings; where researchers can view event reports; and administrators can edit event report data sets and make changes to event data.

1.2.1. Event Reporting

Ordinary citizens will be able to report events that can then be viewed by researchers and academics.

1.2.2. Event Set Editing

Administrators can edit existing events sets to refine what users report, and create new event sets to allow users to start reporting events.

1.2.3. Event Viewing

Researchers will be able to view events in a map representation. This project may be extended to include database queries about the event sets.

1.3. Systems Goals

The goal of this system is to fulfill the purpose of this system, and create a generic crowdsourcing webstie.

1.4. System Overview and Diagram

The webpage will consist of the major components. The main page will contain a map and detailed list of events for the selected event set. There will be a button on the main page that will take the user to a reporting interface. The user can login, and if the user is an administrator, the main page will contain buttons leading to the pages for editing and creating event sets and editing individual events. See Figure 1.1..

1.5. Technologies Overview

This section contains a list of specific technologies used to develop the system. See Table 1.1.

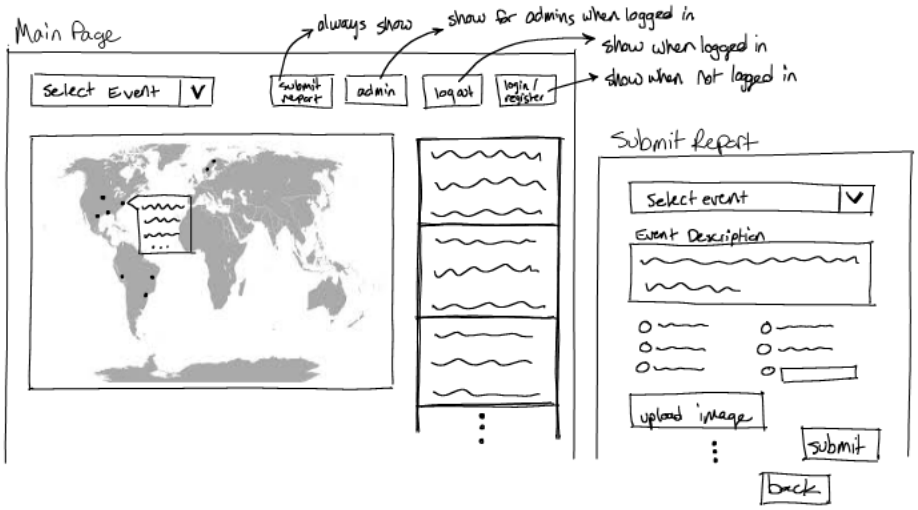

Figure 1.1: Design of Crowd Science Main Page

Table 1.1: Technologies Used

Technology	***Description***	***Reference Material***	***System Usage***
Apache 2.0	*Used for website server hosting.*	`http://httpd.apache.org/docs/2.0/`	*Used to host Crowd Science Mapper website.*
HTML	*Hypertext Markup Language, basic webpage scripting language.*	`http://html.net/`	*Used to tie together JavaScript, PHP, and CSS webpages.*
JavaScript	*More advanced language to create more complex objects.*	`http://html.net/`	*Used to create more complex objects, such as the event map.*
PHP	*Hypertext Preprocesser, used with HTML to provide stateful webpages.*	`http://html.net/`	*Used to send and recieve information from the Mongo Database.*
CSS	*Cascading Style Sheet, used to create a unified look and feel for websites.*	`http://html.net/`	*Used to create look and feel of Crowd Science Mapper.*
Mongo Database	*Used to store information.*	`http://www.mongodb.org/`	*Used to store user login information and event reports.*

2. Project Overview

This section provides information about the team members, team member roles, project management approach and phase overview.

2.1. Team Members and Roles

Team members are Hannah Aker, in the role of Scrum Master, and Jiasong Yan, in the role of team member.

2.2. Project Management Approach

The project will be managed using an Agile approach. There will be 6 sprints total in this project, each lasting 3 weeks. The project backlog is owned by the team, and located on the team Github repository. All parties have access to the Sprint and Product Backlogs. Github issues will be used to track sprint tasks, bugs or trouble tickets, and user stories. The github repository is located at: `https://github.com/SDSMT-CSC464-F15/crowdscience`

2.3. Phase Overview

This project will be implemented in two phases. The first phase will entail analyzing the Landscape Change Mapper project and adapting a copy of the project for expansion into a generic Crowd Science Mapper. The second phase will entail adding the necessary freatures to the base project from phase one to create a proof of concept website.

Within phase one, we will implement and then test the adapted Landscape Change Mapper features we add to the project. Delivery is incorporated into the texting phase, because all testing will be done on the live server. Phase two will also include a design phase where we fully design the new features to be added.

2.4. Terminology and Acronyms

SGT: Stinger Ghaffarian Technologies LCM: Landscape Change Mapper HTML: Hyper-Text Markup Language CSS: Cascading Style Sheet PHP: Hypertext Preprocessor

3. User Stories, Backlog and Requirements

3.1. Overview

This section contains basic requirements for the Crowd Science Mapper project. The Crowd Science Mapper will be a generic crowdsourcing website, where ordinary citizens can report events such as butterfly or bird sightings, and view these events. Administrators in this system will be able to approve individual events and modify event sets, with no programming experience using a graphical user interface.

3.1.1. Scope

This section contains stakeholder information, initial user stories, requirements, product backlog, and proof of concept results.

3.1.2. Purpose of the System

The purpose of this crowdsourcing website is to provide a customizable interface for researchers and academics to collect data about scientific events from ordinary citizens.

3.2. Stakeholder Information

Stakeholders for this project are academics and researchers who would eventually be the administrators of this project, and ordinary citizens who would be the main users, in the feild reporting events.

3.2.1. Customer or End User (Product Owner)

Product owners are Dr. Mengyu Qiao and Gail Schmidt, who will assist in the project in mentor roles. As needed, they will assist with prioritizing the product backlog, and identify important features. Intellicetual property in this project belongs to South Dakota School of Mines and Technology.

3.2.2. Developers –Testers

Developers and testers are Jiasong Yan and Hannah Aker. Testers may eventually include SDSMT faculty and students, with faculty as administrators and students as ordinary users reporting events.

3.3. Requirements and Design Constraints

This project will be a HTML based webpage using Java Script to connect to a Mongo database, and hosted on an Apache server.

3.3.1. System Requirements

Webpage should be able to run on all operating systems and all internet browsers. This project is not required to have mobile functionality.

3.3.2. Network Requirements

This project will be server based, hosted on an Apache 2.0 server, access supplied to us by Gail Schmidt. The server is a cloud based server operated by SGT. Information about Apache 2.0 servers can be found here: `http://httpd.apache.org/docs/2.0/`.

3.3.3. Development Environment Requirements

Development will be done in HTML, CSS, PHP and Java Script. The project will connect to a Mongo database.`https://www.mongodb.org/`. Information about HTML, CSS, PHP and Javascript can be found here: `http://html.net/`.

3.3.4. Project Management Methodology

Source control will be Github; github issues will be used to keep track of backlog and sprint status. The github repository is located at: `https://github.com/SDSMT-CSC464-F15/crowdscience` All parties have access to the Sprint and Product Backlogs. There will be 6 sprints total in this project, each lasting 3 weeks.

3.4. User Stories

3.4.1. User Story #1

As a user, I want all the core functionality of the landscape change mapper to be maintained. These functions include, but are not limited to:

- Visual map representation of events
- Detailed event list
- Event reporting interface
- New user registration
- User login

3.4.2. User Story #2

As an administrator, I want to be able to customize a set of events to fit my needs. This may include user input and display items, database design, digital map, webpage color and style, logo, etc.

3.4.3. User Story #3

As an administrator, I want to be able to edit existing event reports.

3.4.4. User Story #4

As a user, I want to be able to select which set of events I would like to view.

3.5. Requirements

The requirements are directly derived from the user stories, but much more refined. These requirements will map directly to test cases in a tracibility matrix.

3.5.1. Global

- Each page shall contain a navigation bar at the top of the page.
- The navigation bar will have buttons for logging in, new user registration, administration, and event reporting.
- Each page shall contain a drop down box for selecting the event set.

3.5.2. New User Registration

- When no user is logged in, there will be a button on the main page leading to a new user registration page.
- The registration page will have a feild for username, password and password verification, and a “submit” button.
- If the passwords do not match, the user will be notified, and need to reenter one or both passwords.
- After registering, a user will automatically be logged in.

3.5.3. User Login

- When no user is logged in, there will be a button on the main page leading to a login page
- The login page will have fields for entering username and passwords, a “login” button and a link to the user registration page.
- The user will be notified if the username or password was incorrect, and may need to reenter one or both feilds.
- When the user is logged in, an icon indicating the logged in user and a log out button will replace the login button on the navigation page.

3.5.4. Event reporting interface

- When a user is logged in, the user is listed as the author of the report.
- When no user is logged in, the author of the report is “Annonymous”.
- Every event report shall include a longitude and latitude feild.
- The event reporting window shall contain the feilds specified for that event data set.
- Each required feild will be marked as required.
- The event reporting window shall contain a submit button.
- The user will be notified if they have not entered all required feilds, and will be able to return to edit their unfinished report.

3.5.5. Detailed event list

- The detailed event list will be a table of the current event set, and will include all feilds specified for that event data set.
- Each entry shall include longitude, latitude of event, and the author of the event.
- When a point on the map is hovered over, the associated entry in the list will be highlighted.
- When a point on the map is clicked, the associated entry in the list will be highlighted in a different shade.

3.5.6. Map representation of events

- The map shall be a visual representation of the event data set.
- The map shall contain a marker for each event report, placed at the longitude and latitude specified in the report.
- The map will be able to be zoomed in, zoomed out, and panned.
- Entries in the event list will be highlighted when a marker is hovered over or clicked.

3.5.7. Administrator Login

- When an administrator logs in, they wil lbe able to see buttons in the navigation bar that lead to pages for editing events and customizing event sets.

3.5.8. Administrator editing of existing events

- The detailed data list shall be displayed in a seperate window.
- Each event report in the list will have an "edit" button.
- When an administrator clicks the "edit" button, they will be taken to the event reporting interface, with the added feild "Reason for editing".
- When an administrator edits an event report, the administrator name, reason for editing, and date and time shall be recorded and added to the event's history.
- An administrator may mark an event report as false, the administrator name, reason for declaring false, and date and time shall be added to the event's history.

3.5.9. Administrator customization of event sets

- The administrator will be able to create a new event data set.
- The administrator will be able to edit the details of an existing event data set.
- When the administrator clicks on the new event data set, a blank event data set will be created.
- The display name and identification name of an event data set cannot be empty or null.
- The identification name will be used to reference the database, and cannot be changed after creation.
- The administrator will be able to add feilds to their event data set by clicking an "Add new Feild" button.
- The administrator will be able to designate the feild identification name, display name, data type, display method if appliciable, and acceptable values if appliciable.
- If the administrator wants to change the data type of the feild, they will need to remove the current feild and recreate it with the new data type, and data stored in the previous type will be lost.
- The administrator will be able to edit display name, display method, and acceptable values with ease.
- The data types that can be selected will be number, short text, long text, radio button group, check box group, and picture upload.
- When the administrator is finished editing or creating their event data set, they will be able to click "save" to save their data.
- If there are problems with what the administrator has entered, the administrator will be notified and prompted to fix the errors.

3.5.10. User selection of data to view

- User will beable to select from a drop down menu on any page which event data set to view.
- When the current event data set is changed, the map, event list, administrator functions and report features will change according to the feilds specified for that set.
- If implemented, the general appearance of the pages will change according to the selected event data set.

4. Design and Implementation

This section is used to describe the design details for each of the major components in the system. Note that this chapter is critical for all tracks. Research tracks would do experimental design here where other tracks would include the engineering design aspects. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code. Sample algorithm: Algorithm 1. This algorithm environment is automatically placed - meaning it floats. You don't have to worry about placement or numbering.

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y \Leftarrow 1$
  if $n < 0$ **then**
    $X \Leftarrow 1/x$
    $N \Leftarrow -n$
  else
    $X \Leftarrow x$
    $N \Leftarrow n$
  end if
  while $N \neq 0$ **do**
    if N is even **then**
      $X \Leftarrow X \times X$
      $N \Leftarrow N/2$
    else {N is odd}
      $y \Leftarrow y \times X$
      $N \Leftarrow N - 1$
    end if
  end while

Citations look like [2, 1, 3] and [6, 4, 5]. These are done automatically. Just fill in the database `designrefs.bib` using the same field structure as the other entries. Then pdflatex the document, bibtex the document and pdflatex twice again. The first pdflatex creates requests for bibliography entries. The bibtex extracts and formats the requested entries. The next pdflatex puts them in order and assigns labels. The final pdflatex replaces references in the text with the assigned labels. The bibliography is automatically constructed.

4.1. Major Component #1

4.1.1. Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.1.2. Component Overview

This section can take the form of a list of features.

4.1.3. Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.1.4. Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.1.5. Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.1.6. Design Details

This is where the details are presented and may contain subsections. Here is an example code listing:

```
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;

    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
```

This code listing is not floating or automatically numbered. If you want auto-numbering, but it in the algorithm environment (not algorithmic however) shown above.

4.2. Major Component #2

4.2.1. Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.2.2. Component Overview

This section can take the form of a list of features.

4.2.3. Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.2.4. Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.2.5. Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.2.6. Design Details

This is where the details are presented and may contain subsections.

4.3. Major Component #3

4.3.1. Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.3.2. Component Overview

This section can take the form of a list of features.

4.3.3. Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.3.4. Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.3.5. Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.3.6. Design Details

This is where the details are presented and may contain subsections.

5. System and Unit Testing

This section describes the approach taken with regard to system and unit testing.

5.1. Overview

Provides a brief overview of the testing approach, testing frameworks, and general how testing is/will be done to provide a measure of success for the system.

5.2. Dependencies

Describe the basic dependencies which should include unit testing frameworks and reference material.

5.3. Test Setup and Execution

Describe how test cases were developed, setup, and executed. This section can be extremely involved if a complete list of test cases was warranted for the system.

6. Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

6.1. Development IDE and Tools

Describe which IDE and provide links to installs and/or reference material.

6.2. Source Control

Which source control system is/was used? How was it setup? How does a developer connect to it?

6.3. Dependencies

Describe all dependencies associated with developing the system.

6.4. Build Environment

How are the packages built? Are there build scripts?

6.5. Development Machine Setup

If warranted, provide a list of steps and details associated with setting up a machine for use by a developer.

7. User Documentation

This section should contain the basis for any end user documentation for the system. End user documentation would cover the basic steps for setup and use of the system. It is likely that the majority of this section would be present in its own document to be delivered to the end user. However, it is recommended the original is contained and maintained in this document.

7.1. User Guide

The source for the user guide can go here. You have some options for how to handle the user docs. If you have some `newpage` commands around the guide then you can just print out those pages. If a different formatting is required, then have the source in a separate file `userguide.tex` and include that file here. That file can also be included into a driver (like the senior design template) which has the client specified formatting. Again, this is a single source approach.

7.2. Installation Guide

7.3. Programmer Manual

8. Class Index

8.1. Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

9. Class Documentation

9.1. Poly Class Reference

Public Member Functions

- Poly ()
- ∼Poly ()
- int myfunction (int)

9.1.1. Constructor & Destructor Documentation

9.1.1.a Poly::Poly ( )

My constructor

9.1.1.b Poly::∼Poly ( )

My destructor

9.1.2. Member Function Documentation

9.1.2.a int Poly::myfunction (int *a*)

my own example function fancy new function
new variable
The documentation for this class was generated from the following file:

- hello.cpp

Bibliography

[1] R. Arkin. *Governing Lethal Behavior in Autonomous Robots*. Taylor & Francis, 2009.

[2] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, June 2005.

[3] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[4] V. Lumelsky and A. Stepanov. Path planning strategies for point mobile automation moving amidst unknown obstacles of arbirary shape. *Algorithmica*, pages 403–430, 1987.

[5] S.A. NOLFI and D.A. FLOREANO. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. A Bradford book. A BRADFORD BOOK/THE MIT PRESS, 2000.

[6] Wikipedia. Asimo — Wikipedia, the free encyclopedia. `http://upload.wikimedia.org/wikipedia/commons/thumb/0/05/HONDA_ASIMO.jpg/450px-HONDA_ASIMO.jpg`, 2013. [Online; accessed June 23, 2013].

SDSMT SENIOR DESIGN
SOFTWARE DEVELOPMENT AGREEMENT

This Software Development Agreement (the "Agreement") is made between the SDSMT Computer Science

Senior Design Team _____,
("Student Group")

consisting of team members _____,
("Student Names")

and Sponsor _____,
("Company Name")

with address: _____.

[Note: Bracketed material is included to suggest content that will vary with each agreement. I STRONGLY SUGGEST THAT THE INSTRUCTOR LOOK AT THE COMPLETED AGREEMENT BEFORE YOU SIGN IT!!]

1 RECITALS

1. Sponsor desires Senior Design Team to develop software [for use in Sponsor's simulation platform for optical fiber transmissions of digitized video signals] (the "Field").
2. Senior Design Teams willing to develop such Software.

NOW, THEREFORE, in consideration of the mutual covenants and promises herein contained, the Team and Sponsor agree as follows:

2 EFFECTIVE DATE

This Agreement shall be effective as of _____ (the "Effective Date").

3 DEFINITIONS

1. "Software" shall mean [the computer programs in machine readable object code form and any subsequent error corrections or updates supplied to Sponsor by Senior Design Team pursuant to this Agreement.] [Depending on the particulars of each agreement, any or all of the following may need to be specified. If they are relevant, they should be used throughout, modifying the standard form as appropriate.]
2. "Acceptance Criteria" means the written technical and operational performance and functional criteria and documentation standards set out in the [project plan.]
3. "Acceptance Date" means [the date for each Milestone when all Deliverables included in that Milestone have been accepted by Sponsor in accordance with the Acceptance Criteria and this Agreement.]
4. "Deliverable" means a deliverable specified in the [project plan.]
5. "Delivery Date" shall mean, [with respect to a particular Milestone,] the date on which University has delivered to Sponsor all of the Deliverables [for that Milestone] in accordance with [the project plan and] this Agreement.

6. "Documentation" means the documents, manuals and written materials (including end-user manuals) referenced, indicated or described in [the project plan] or otherwise developed pursuant to this Agreement.
7. "Milestone" means the completion and delivery of all of the Deliverables or other events which are included or described in [the project plan] scheduled for delivery and/or completion on a given target date; a Milestone will not be considered completed until the Acceptance Date has occurred with respect to all of the Deliverables for that Milestone.

4 DEVELOPMENT OF SOFTWARE

1. Senior Design Team will use its best efforts to develop the Software described in [the project plan.] The Software development will be under the direction of or his/her successors as mutually agreed to by the parties ("Team Lead") and will be conducted by the Team Lead. The Team will deliver the Software to the satisfaction of the course instructor that reasonable effort has been made to address the needs of the client. The Team understands that failure to deliver the Software is grounds for failing the course.
2. Sponsor understands that the Senior Design course's mission is education and advancement of knowledge, and, consequently, the development of Software must further that mission. The Senior Design Course does not guarantee specific results or any results, and the Software will be developed only on a best efforts basis. The Software is considered PROOF OF CONCEPT only and is NOT intended for commercial, medical, mission critical or industrial applications.
3. The Senior Design instructor will act as mediator between Sponsor and Team; and resolve any conflicts that may arise.

5 COMPENSATION

[This is entirely subject to negotiation. Normally NO COMPENSATION occurs in a Senior Design Project. On occasion an intern status and wage is appropriate.]

6 CONSULTATION AND REPORTS

1. Sponsor's designated representative for consultation and communications with the Team Lead shall be

   _____ or such other person as Sponsor may from time to time designate to the Team Lead ("Designated Representative").
2. During the Term of the Agreement, Sponsor's representatives may consult informally with course instructor regarding the project, both personally and by telephone. Access to work carried on in University facilities, if any, in the course of this Agreement shall be entirely under the control of University personnel but shall be made available on a reasonable basis.
3. The Team Lead will submit written progress reports. At the conclusion of this Agreement, the Team Lead shall submit a comprehensive final report in the form of the formal course documentation at the conclusion of the Senior Design II course.

7 CONFIDENTIAL INFORMATION

1. The parties may wish, from time to time, in connection with work contemplated under this Agreement, to disclose confidential information to each other ("Confidential Information"). Each party will use reasonable efforts to prevent the disclosure of any of the other party's Confidential Information to third parties for

a period of three (3) years after the termination of this Agreement, provided that the recipient party's obligation shall not apply to information that:

(a) is not disclosed in writing or reduced to writing and so marked with an appropriate confidentiality legend within thirty (30) days of disclosure;

(b) is already in the recipient party's possession at the time of disclosure thereof;

(c) is or later becomes part of the public domain through no fault of the recipient party;

(d) is received from a third party having no obligations of confidentiality to the disclosing party;

(e) is independently developed by the recipient party; or

(f) is required by law or regulation to be disclosed.

2. In the event that information is required to be disclosed pursuant to subsection (6), the party required to make disclosure shall notify the other to allow that party to assert whatever exclusions or exemptions may be available to it under such law or regulation.

8 INTELLECTUAL PROPERTY RIGHTS

[Negotiated on a case-by-case basis. This must address who owns the algorithms and who owns the source code. For example: All deliverables become property of the Sponsor. Roughly: If the idea originates with the sponsor, or if a sponsor pays you to develop an idea, then they have legitimate claim to the IP. If the idea originates from the University (through faculty or staff) then the University has legitimate claim. If the idea is yours (student) and you develop it without external compensation then you have legitimate claim.]

9 WARRANTIES

The Senior Design Team represents and warrants to Sponsor that:

1. the Software is the original work of the Senior Design Team in each and all aspects;
2. the Software and its use do not infringe any copyright or trade secret rights of any third party.

No agreements will be made beyond items (1) and (2).

10 INDEMNITY

1. Sponsor is responsible for claims and damages, losses or expenses held against the Sponsor. [Sponsor may have something to add here.]
2. Sponsor shall indemnify and hold harmless the Senior Design Team, its affiliated companies and the officers, agents, directors and employees of the same from any and all claims and damages, losses or expenses, including attorney's fees, caused by any negligent act of Sponsor or any of Sponsor's agents, employees, subcontractors, or suppliers.
3. NEITHER PARTY TO THIS AGREEMENT NOR THEIR AFFILIATED COMPANIES, NOR THE OFFICERS, AGENTS, STUDENTS AND EMPLOYEES OF ANY OF THE FOREGOING, SHALL BE LIABLE TO ANY OTHER PARTY HERETO IN ANY ACTION OR CLAIM FOR CONSEQUENTIAL OR SPECIAL DAMAGES, LOSS OF PROFITS, LOSS OF OPPORTUNITY, LOSS OF PRODUCT OR LOSS OF USE, WHETHER THE ACTION IN WHICH RECOVERY OF DAMAGES IS SOUGHT IS BASED ON CONTRACT TORT (INCLUDING SOLE, CONCURRENT OR OTHER NEGLIGENCE AND STRICT

LIABILITY), STATUTE OR OTHERWISE. TO THE EXTENT PERMITTED BY LAW, ANY STATUTORY REMEDIES WHICH ARE INCONSISTENT WITH THE PROVISIONS OF THESE TERMS ARE WAIVED.

11 INDEPENDENT CONTRACTOR

For the purposes of this Agreement and all services to be provided hereunder, the parties shall be, and shall be deemed to be, independent contractors and not agents or employees of the other party. Neither party shall have authority to make any statements, representations or commitments of any kind, or to take any action which shall be binding on the other party, except as may be expressly provided for herein or authorized in writing.

12 TERM AND TERMINATION

1. This Agreement shall commence on the Effective Date and extend until the end of classes of the second semester of Senior Design (CSC 467), unless sooner terminated in accordance with the provisions of this Section ("Term").
2. This Agreement may be terminated by the written agreement of both parties.
3. In the event that either party shall be in default of its materials obligations under this Agreement and shall fail to remedy such default within thirty (30) days after receipt of written notice thereof, this Agreement shall terminate upon expiration of the thirty (30) day period.
4. Any provisions of this Agreement which by their nature extend beyond termination shall survive such termination.

13 ATTACHMENTS

Attachments A and B are incorporated and made a part of this Agreement for all purposes.

14 GENERAL

1. This Agreement constitutes the entire and only agreement between the parties relating to the Senior Design Course, and all prior negotiations, representations, agreements and understandings are superseded hereby. No agreements altering or supplementing the terms hereof may be made except by means of a written document signed by the duly authorized representatives of the parties.
2. This Agreement shall be governed by, construed, and enforced in accordance with the internal laws of the State of South Dakota.

15 SIGNATURES

Replace with name of student #1	Date
Replace with name of student #2	Date
Replace with name of student #3	Date
Replace with name of sponsor's representative	Date

A. Product Description

Write a description of the product to be developed. Use sectioning commands as neccessary.

NOTE: *This is part of the contract.*

B. Sprint Reports

1. Sprint Report #1

1.1. Team Members

Hannah Aker and Jiasong Yan

1.2. Project Sponsors

Dr. Mengyu Qiao and Gail Schmidt

1.3. Sponsor/User Description

1.3.a User Description

Primary users will be the everyday citizen, interested in reporting some event, such as butterfly sighting, geological or landscape changes, etc. Secondary users will be academics and researchers who will use the gathered information in their research, they will be administrators of this data.

1.3.b Project Goal

The goal of this project to improve on the idea originally presented in the Landscape Change Mapper, and expand on it to create a flexible interface for other kinds of events.

1.3.c User Needs

Primary users need all the core fonctionality of the Landscape Change Mapper to be maintained.These functions include, but are not limited to:

- Visual map representation of events
- Detailed event list
- Event reporting interface
- New user registration
- User login

Users also need to be able to select which set of events they would like to view. Administrators need to be able to customize a set of events to fit my needs. This may include user input and display items, database design, digital map, webpage color and style, logo, etc. Administrators will also need to be able to edit existing event reports.

1.4. Project Overview

The Crowd Science Mapper will be a generic crowdsourcing system framework and toolkits, which can be customized by ordinary users with no programming experience using graphical user interface

1.5. Project Environment

1.5.a Project Boundaries

While the previous project included a mobile application, this project will not because the team is smaller. This project will be solely web-based.

1.5.b Project Context

This project will use the same general context that the Landscape Change Mapper used.

1.5.c Technical Environment

This project will use the same enviroment used in the Landscape Change Mapper. This will use HTML, PHP and Java Script to connect to a Mongo Database, and was hosted on an Apache 2.2 Server.

1.5.d Current systems overview

The Landscape Change Mapper webpages used HTML, PHP and Java Script to connect to a Mongo Database, and was hosted on an Apache 2.2 Server. The webpage contained a map of events, detailed event list, event reporting interface, user login, and new user registration. The project included a mobile app with an event reporting interface.

1.6. Project Deliverables

The project deliverable will be a proof-of-concept webpage with the features listed in the backlog.

1.7. Backlog

The following features need to be added to this project:

- Visual map representation of events
- Detailed event list
- Event reporting interface
- New user registration
- User login
- User selection of data set to view
- Administrator login
- Administrator customization of event sets, including user input and display items, database design, digital map, webpage color and style, logo, etc.
- Administrator editing of existing event reports

1.8. Potential Issues

Potential issues might stem from using Java Script to interface with the Mongo DB. We don't have much prior experience with these specific tools, though we have used similar tools.

2. Sprint Report #2

3. Sprint Report #3

4. Sprint Report ...

C. Industrial Experience and Resumes

1. Resumes

Your resumes are included here. See the source file (industrial.tex) and uncomment the PDF includes to see how this works. If your resume is written in LaTeX then you can just insert the LaTeX source code.

2. ABET: Industrial Experience Reports

2.1. Name1

2.2. Name2

2.3. Name3

D. Acknowledgment

Thanks

E. Supporting Materials

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.