

# Hotspots – Design Specification

Ryan Coe - Brittany Kos - Jes Stockhausen

## Visual Design

Each of the pages will have a similar look and feel, keeping cluster to a minimal. We will use the *include* function in the header to use the same CSS for each page. Adobe Kuler will provide a color palette for the basic design of the site. We have chosen to utilize the Seaside color palate: <http://kuler.adobe.com/#themes/search?term=userId%3A144382>

## General

- Home
  - The home page will be very simple. The Hotspot logo will be displayed in the top left corner as well as limited featured Hotspots displayed below. There will be a login field available in the top right hand corner of the page with two text fields; a user name and password box.
- New Account Page
  - The new account page will simply consist of the Hotspot header and navigation bar with two text fields. One for available username and one for the desired password.
- Hotspot Page
  - Displays one single hotspot. There will be multiple picture sat the top of the page as well as a description and comments from users in a div below.

## User

- User Home
  - User home will be one of the more full pages. It will consist of a short welcome message, a view road trip tab, request hotspot tab, recent hotspots, as well as the hotspots logo and navigation bar.
- View Trip Page
  - View trip page will have a Google API that maps out the user's planned trip. as well as the hotspots logo and navigation bar.
- Request Hotspot Page
  - This page will be a simple send a comment to the admins. It will consist of A text field for the users name as and another for their comment. as well as the hotspots logo and navigation bar at the top of the page.

## Admin

- Admin Home
  - Will consist of a short welcome message, as well as the hotspots logo and navigation bar.
- Add Hotspot (admin)
  - Will allow the admin to add new hotspots to the hotspot field. The page will be simple. It will allow the admin to add a photo, description, and a short comment.
- Edit Hotspots page
  - The Hotspot page will give the admin the opportunity to edit the hotspot visual page. It will be very similar in looks and the admin will have the ability to edit text and move thing around.

## **Program Flow**

A visitor to the site will first land on the Home page. This page will show some featured hotspots and an option for returning users to log in. If the user clicks on a featured hotspot, they will be directed to that hotspot's page. If they log in, they will be sent to their User Home. This same login field will also be used by administrators, but they will be sent to the Admin Home. The User Home will also provide a link to the Request to Add Hotspot page, where users can suggest new hotspots. If logged in as an admin, then you can click on a link to the Add Hotspot page or the Edit Hotspot page. Users will also have an option to click on a link to view the various hotspots.

## **Pseudocode - Level 1**

### **General:**

#### **Home (Brit)**

featured hotspots -> hotspot page

login -> login page

create account -> create account page

hotspots navigation -> click on hotspot, go to hotspots page

#### **New Account Page (Jes)**

form with user name, password, and password

submit button -> validate input

if user name taken, or passwords the same will print error in page

link to user home

#### **Hotspot Page (Ryan)**

shows "hotspot" in general

will have a description of the hotspot, user comments, and thumbnails of photos people have uploaded. The photos will be expanded into a lightbox

### **User:**

#### **User Home (Ryan)**

welcome message

view roadtrip -> View Roadtrip Page

request to add hotspot -> Request hotspot page

create new roadtrip -> go to Roadtrip Page (will not be populated with info yet)

if user already has a road trip, they will not be able to create new roadtrip

#### **View Roadtrip Page (Brit)**

Google API,

list of "hotspots" on route, with delete button next to them

map of us states

click on state, will display hotspots for each state and add button next to each

#### **Request "hotspot" page (Jes)**

form with name, location, description, and send button

after submitted, will take back to user home page

## **Admin:**

### **Admin Home (Jes)**

welcome message  
edit hotspots -> Edit hotspots page  
add hotspots -> add hotspots page  
google analytics link -> external link to google analytics

### **Add "hotspot" (Ryan)**

name, location, description, photos, and user comments

### **Search hotspots (Brit)**

click on specific hotspot -> go to edit hotspots page

### **Edit Hotspots Page (Brit)**

map of hotspots  
click on state and view the hotspots there  
form with name, location, description, and photo links prepopulated in text fields. user will edit text and click save button. This will change text directly in database. Admin will then go back to blank Edit Hotspots Page

## **Pseudocode - Level 2**

### **General:**

#### **Home (Brit/Jes/Ryan)**

hotspots logo prominently displayed  
featured hotspots -> hotspot page  
login div  
    submit button (with proper username and password) will send registered user into the system  
create account -> create account page  
hotspots navigation -> click on hotspot, go to hotspots page

#### **New Account Page (Jes)**

form with user name, password  
get all users names from database and store in \$name  
form with user name, password submit button  
user clicks submit button  
    if \$newuser == \$name  
if user name taken, or passwords the same will print error in page  
    ask user for another user name  
    else add user name and password in database, display link to user home page

#### **Hotspot Page (Ryan)**

shows "hotspot" in general  
will have a description of the hotspot, user comments, and thumbnails of photos people have uploaded. The photos will be expanded into a lightbox  
\*top header/navigation included on page via include css function\*  
hotspot name shown at top along with state  
small thumbnails (200px tall) of hotspot photos shown across the page below name  
    can be expanded via lightbox[rel] function, allowing for scrolling without closing window  
admin description of hotspot from database

user comments below admin description  
form at bottom for submitting user comments/reviews  
submit review -> add comment to database

## **User:**

### **User Home (Ryan)**

welcome message  
view roadtrip link -> View Roadtrip Page  
request to add hotspot link -> Request hotspot page  
create new roadtrip link -> go to Roadtrip Page (will not be populated with info yet)  
if user already has a road trip, they will not be able to create new roadtrip  
only 1 trip can be linked to each user account

### **View Roadtrip Page (Brit)**

- gets \$userID, \$roadtripID, and array of \$hotspots from POST variables
- if no \$roadtripID looks up the road trip id attached to that user id
  - if no roadtrip id found for user, then create new roadtrip for them
- if empty(\$hotspots) looks up the hotspot ids attached to the roadtrip id.
  - if no hotspots are attached to the road trip, leave all hotspots information blank
- checks for \$deletedHotspotID
  - if found, delete that hotspot id from \$hotspots and in database
- checks for \$addedHotspotID
  - if found, add that hotspot id to \$hotspots and in database
- creates google map (in javascript)
  - feeds \$hotspots to google map
- displays google map
- under map, have deleteHotspots div by looping through \$hotspots
  - show brief description and thumbnail
    - link to Hotspots Page if clicked on
  - a "delete" button, each with an individual form that has a self action
    - passed is the \$deletedHotspotID
    - hidden variables are \$userID, \$roadtripID, and array of \$hotspots
- under that div display the USmap div with each state
  - if a state is clicked on, then populate the addHotspots div with the hotspots for that state.
    - if the state does not have hotspots, then display nothing in addHotspots div
  - each hotspot should have a brief description and thumbnails
    - if clicked on, will take to Hotspot Page
  - a "add" button will also be displayed next to each hotspot with individual forms
    - passed is \$addedHotspotID
    - hidden variables will be added to the \$userID, \$roadtripID, and array of \$hotspots

### **Request "hotspot" page (Jes)**

form with name, location, description, and send button  
form <input text> \$name, \$location,  
form <input text field> \$description  
enter to database  
after submitted, will take back to user home page

## **Admin:**

### **Admin Home (Jes)**

welcome message  
get \$userid from POST variable  
form <button> edit hotspots -> Edit Hotspots page

hidden \$userid  
form <button> add hotspots -> Add Hotspots page  
hidden \$userid  
google analytics link -> external link to google analytics

### **Add “hotspot” (Ryan)**

name, location, description, photos, and user comments  
simple form page containing included css like other pages  
text field: name of hotspot  
    \$hotspotname  
text field: state of hotspot  
    \$hotspotstate  
text field: user description  
    \$hotspotdescr  
submit button -> add hotspot to database

### **Edit Hotspots Page (Brit)**

- check if \$editHotspotID is set in \$POST variable
  - edit \$name, \$location, \$description, and \$imgLink[] to that hotspot
- display map of hotspots
- if a state is clicked on, display the state’s hotspots in listHotspots div
- if a hotspot is clicked, display form with name, location, description, and photo links prepopulated in text fields in a editHotspot div
- admin can edit text directly in these text fields
  - form will be sent with \$name, \$location, \$description, and \$imgLink[]
  - hidden will be \$editHotspotID

## **Calendar**

Meetings once a week. We have the work divided up based on page. The following calendar describes what each of us plan to do during each week,

- 4/4 design spec finished, visual design created, each do simple versions of each of our pages
- 4/11 have a functioning database and be able to simple navigate through site
- 4/18 have site functioning, connecting to database
- 4/25, 4/26, 4/27 testing of site

## **Testing**

We will each create our own part of the website and do testing while we build. In each of the weekly meetings we will exchange code and make sure our pages will be able to link up to each other. This is when we will test each other’s pages. Since we will each be creating individual pages and combining them together, the other two team members will act as testers who have never seen the pages before and will test them from scratch. Mondays will also be focused on the hand off between pages and making sure the correct variables are being passed to each other.

The last week of school is also reserved for testing and the three days we will meet will act as updates where we will tell each other where changes need to be made. Those changes will then be made at those meetings and code exchanged again.