

---

CS 148 Database Design for the Web

Alan Chu and Joe Siebert

Final Project

---

UVM diddit

Version <1.0>

Title of project	Version <1.0>
	December 12, 2015

## Table Of Contents

<b>Time Log .....</b>	<b>3</b>
<b>Introduction .....</b>	<b>6</b>
Purpose.....	6
Definitions, acronyms, and abbreviations .....	6
<b>Overall Description .....</b>	<b>6</b>
<b>Data Dictionary .....</b>	<b>7</b>
<b>ER Diagram .....</b>	<b>8</b>
<b>Schema .....</b>	<b>8</b>
<b>Story Board .....</b>	<b>12</b>
<b>Specific requirements .....</b>	<b>13</b>

Title of project	Version <1.0>
	December 12, 2015

Time Log			
Estimated time to complete assignment =>			60 Hours
Date	Time Spent (in hours)	Description	Author
11/3/15	1.0	First version of software requirements	Joe Siebert
11/4/15	0.25	Sketched preliminary storyboard	Joe Siebert
11/5/15	1.5	Created data dictionary; created tables using PHP myAdmin, typed in sample records, and tested functionality of table organization; created .sql files needed to recreate tables	Joe Siebert
11/6/15	1.0	Sized out elements for fluid CSS grid	Joe Siebert
11/8/15	0.25	Adding additional fields to tblActivities	Joe Siebert
11/8/15	0.5	Inserting records in tblActivities, tblTowns, and tblVotes; adjusted top-10 SQL query to grab only the top ten records	Joe Siebert
11/9/15	0.5	Researching logic necessary to update variables for up-/downvoting system	Joe Siebert
11/9/15	0.5	Created form.php; created read-only NetID field on form that gets the username for the UVM authentication system	Joe Siebert
11/9/15	0.75	Added two fields to form.php: activity name and category.	Joe Siebert
11/10/15	1.0	Added several fields to form.php; added an insert statement that inserts data into required fields of tblActivities	Joe Siebert
11/11/15	2.0	Added security feature to check whether users are administrators; began admin.php, adding the first feature: an interface for admins to approve submitted records	Joe Siebert
11/11/15	1.25	Added basic voting system to top-10.php	Joe Siebert
11/13/15	1.0	Changed voting system to use buttons; added Edit links to top.php	Joe Siebert
11/15/15	2.0	Updated form.php to allow administrators to	Joe Siebert

Title of project	Version <1.0>
	December 12, 2015

		edit records	
11/16/15	1.0	Updated form.php to add relational records when necessary (i.e., if town of activity is not yet in the town tables)	Joe Siebert
11/17/15	1.5	Updated form.php with optional fields for Activities table	Joe Siebert
11/18/15	0.25	Updated top-10.php to add users to affiliates table in they're not already in it	Joe Siebert
11/18/15	0.5	Changed adminCheck function in security.php to reference administrators table (vs. hardcoded admin IDs)	Joe Siebert
11/18/15	1.25	Added admin folder and updated filepaths in top.php to accommodate the files in the folder	Joe Siebert
11/19/15	2.5	Added jQuery function to show/hide additional information about the activities	Joe Siebert
11/20/15	1.0	Integrating remove.php into site	Joe Siebert
11/30/15	0.75	Sorting out a merge conflict; adding link to remove.php to top-10.php; fixed validation error	Joe Siebert
11/30/15	0.5	Updating specific requirements section of this document; making minor tweaks to approve.php and top-10.php	Joe Siebert
11/30/15	0.75	Added admin-only checkbox to form.php	Joe Siebert
12/1/15	1.0	Added fldAffiliation to tblAffiliates; updated form.php to include a radio button to add this information to the table; updated top-10.php vote system so that user's affiliate status will be updated with vote	Joe Siebert
12/2/15	1.0	Updated remove.php to display a list of all activities when one is not appended to the URL, and also to show all records (primary and relational) that will be removed if user confirms removal	Joe Siebert
12/2/15	1.5	Researching Zurb Foundation; applying Foundation classes to header and nav	Joe Siebert
12/4/15	3.0	Updated CSS using Foundation; fixed bug in form.php code; changed SQL query in list.php (changed from top-10.php) to grab only ten records at a time; added "Previous" and "Next" links to list.php, allowing users to cycle through ten records at a time; incorporated	Joe Siebert

Title of project	Version <1.0>
	December 12, 2015

		Alan's about.php code into Foundation layout	
12/5/15	1.0	Tweaking CSS style; fixing bug in form.php; incorporated Alan's index.php page	Joe Siebert
12/6/15	1.0	Updated header.php and nav.php to change format of image and nav bar	Joe Siebert
12/6/15	2.0	Incorporated Alan's submit.php code; creating tblPhotos; adding thumbnails to list.php	Joe Siebert
12/7/15	2.5	Incorporated Alan's images.php code; fixing validation errors; updating remove.php	Joe Siebert
12/7/15	1.0	Updating images.php to remove images	Joe Siebert
12/7/15	1.5	Finalizing this document	Joe Siebert
11/15/15	1.5	Get up to speed to Joe's code, forked the repository	Alan Chu
11/15/15	0.5	Tested form and debugged error	Alan Chu
11/25/15	1.5	Structured and added content to about.php	Alan Chu
11/26/15	2.0	Researched flex boxes for about.php and the site generally	Alan Chu
12/1/15	2.0	Created index.php; wrote SQL query to retrieve most recent activities	Alan Chu
12/2/15	1.0	Research Foundation and CSS frameworks	Alan Chu
12/4/15	2.0	Researched image validation	Alan Chu
12/5/15	5.0	Created, tested, and debugged submit.php	Alan Chu
12/7/15	5.0	Finished image upload form and created image.php to approve uploaded images	Alan Chu
	59.5	<=Total Time Spent	

Title of project	Version <1.0>
	December 12, 2015

## Introduction

### Purpose

The purpose of this document is to describe the requirement specification for the website "UVM diddit."

The intended audience for this document is twofold. First, it is the contract between the client and us. Second, it is for the developers of the site.

The two primary goals are:

1. *Client gets a very good understanding of what we will be delivering.*
2. *Developers have a very good understanding of what they are going to create.*

### Definitions, acronyms, and abbreviations

*Sample for the client:*

HTML – Hypertext markup language – used to define your content.

PHP – Personal Home Page – language that helps to customize html.

CSS – Cascading Style Sheets – used to define the look of a web site.

W3 Validation – refers to both Html and CSS validation tool provided by the W3c.org. the html validator is located at:

<http://validator.w3.org/>

with the CSS validator located at:

<http://jigsaw.w3.org/css-validator/>

## Overall Description

UVM diddit will be a site for current and former UVM students to suggest activities that they feel every UVM student should do before they graduate. It was inspired by a list called "101 Things to Do Before You Graduate from UVM," which is distributed to first-year students when they move onto campus. The list is fine and good, but we thought transforming the list into an interactive website would reveal some new activities that were, for one reason or another, left off the university's "official" list. It will also change the list into a ranking of activities, allowing users to vote up or down activities listed on the site.

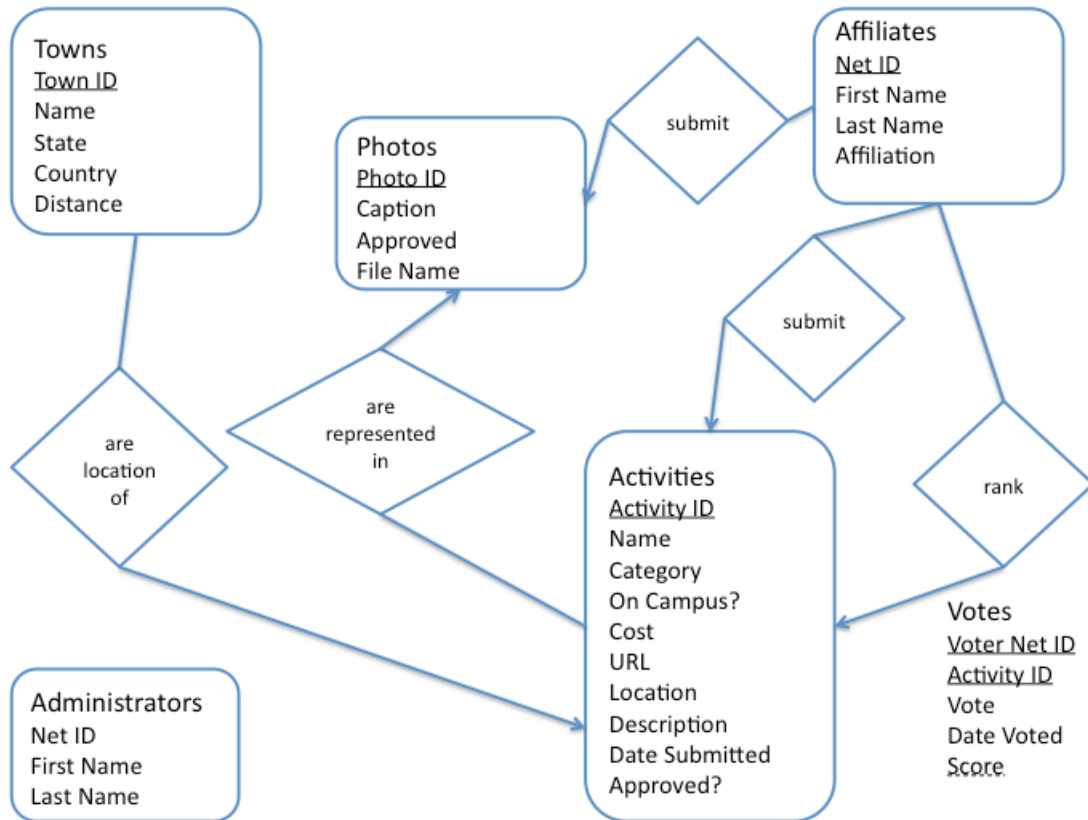
Title of project	Version <1.0>
	December 12, 2015

## Data Dictionary

TABLE	COLUMN	TYPE	PREC	PK	AUTO	DESCRIPTION
Activities	Activity ID	int	11	Y	INCR	Auto number for activity
Activities	Name	varchar	255			Name of activity
Activities	Category	varchar	100			Type of activity
Activities	On Campus	tinyint	1			0/1 to represent true or false
Activities	Cost*	int	12			Number representing cost (in dollars)
Activities	URL*	varchar	255			URL associated with activity
Activities	Location*	varchar	255			Detailed description of location beyond town name
Activities	Description*	text				Comment on activity
Activities	Submitter Net ID	varchar	12			UVM Net ID of user that submitted activity
Activities	Date Submitted	timestamp				Timestamp of activity submission
Activities	Approved	tinyint	1			0/1 to indicate whether activity is approved
Admin	Net ID	varchar	12	Y		UVM Net ID of admin
Admin	First Name*	varchar	255			First name of administrator
Admin	Last Name*	varchar	255			Last name of administrator
Photos	Photo ID	int	11	Y	INCR	Auto number for new photos
Photos	Caption	varchar	256			Caption describing user's image
Photos	Approved	tinyint	1			0/1 to indicate whether photo is approved
Photos	File Name	varchar	256			Filename of image for easy reference in HTML code
Towns	Town ID	int	11	Y	INCR	Auto number for new towns
Towns	Town Name	varchar	100			Name of activity's town
Towns	State	char	2			Abbreviation of state
Towns	Distance	int	11			Mileage from Burlington
Votes	Vote	tinyint	1			Can have three values: -1, 0, 1
Votes	Date Voted	timestamp				Timestamp of most recent vote

Title of project	Version <1.0>
	December 12, 2015

## ER Diagram



## Schema

```
-- Database: `JSIEBERT_ACTIVITIES`
```

```
-- -----
```

```
--
```

```
-- Table structure for table `tblActivities`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tblActivities` (
  `pmkActivityId` int(11) NOT NULL,
  `fldName` varchar(255) NOT NULL,
  `fldCategory` varchar(100) NOT NULL,
  `fldOnCampus` tinyint(1) NOT NULL,
  `fnkTownId` int(11) NOT NULL,
  `fldLocation` varchar(255) DEFAULT NULL,
  `fldCost` int(12) DEFAULT NULL,
  `fldURL` varchar(255) DEFAULT NULL,
```



Title of project	Version <1.0>
	December 12, 2015

```

    `fldDescription` text,
    `fnkSubmitNetId` varchar(12) NOT NULL,
    `fldDateSubmitted` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
    `fldApproved` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=57 DEFAULT CHARSET=utf8;

```

-- -----

```

--
-- Table structure for table `tblAdministrators`
--

```

```

CREATE TABLE IF NOT EXISTS `tblAdministrators` (
  `pmkNetId` varchar(12) NOT NULL,
  `fldFirstName` varchar(255) NOT NULL,
  `fldLastName` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-- -----

```

--
-- Table structure for table `tblAffiliates`
--

```

```

CREATE TABLE IF NOT EXISTS `tblAffiliates` (
  `pmkNetId` varchar(12) NOT NULL,
  `fldAffiliation` varchar(255) NOT NULL DEFAULT 'Other',
  `fldLastName` varchar(100) DEFAULT NULL,
  `fldFirstName` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-- -----

```

--
-- Table structure for table `tblPhotos`
--

```

```

CREATE TABLE IF NOT EXISTS `tblPhotos` (
  `pmkPhotoId` int(11) NOT NULL,
  `fnkActivityId` int(11) NOT NULL,
  `fnkNetId` varchar(12) NOT NULL,
  `fldCaption` varchar(256) NOT NULL,
  `fldApproved` tinyint(1) NOT NULL DEFAULT '0',
  `fldFileName` varchar(256) NOT NULL

```

Title of project	Version <1.0>
	December 12, 2015

```
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=utf8;
```

```
-- -----
```

```
--
-- Table structure for table `tblTowns`
--
```

```
CREATE TABLE IF NOT EXISTS `tblTowns` (
  `pmkTownId` int(11) NOT NULL,
  `fldTownName` varchar(100) NOT NULL,
  `fldState` char(2) NOT NULL,
  `fldDistance` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
```

```
-- -----
```

```
--
-- Table structure for table `tblVotes`
--
```

```
CREATE TABLE IF NOT EXISTS `tblVotes` (
  `fnkNetId` varchar(12) NOT NULL,
  `fnkActivityId` varchar(255) NOT NULL,
  `fldVote` tinyint(1) NOT NULL DEFAULT '0',
  `fldDateVoted` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Indexes for dumped tables
--
```

```
--
-- Indexes for table `tblActivities`
--
ALTER TABLE `tblActivities`
  ADD PRIMARY KEY (`pmkActivityId`);
```

```
--
-- Indexes for table `tblAdministrators`
--
ALTER TABLE `tblAdministrators`
  ADD PRIMARY KEY (`pmkNetId`);
```

```
--
```

Title of project	Version <1.0>
	December 12, 2015

```

-- Indexes for table `tblAffiliates`
--
ALTER TABLE `tblAffiliates`
  ADD PRIMARY KEY (`pmkNetId`);

--
-- Indexes for table `tblPhotos`
--
ALTER TABLE `tblPhotos`
  ADD PRIMARY KEY (`pmkPhotoId`);

--
-- Indexes for table `tblTowns`
--
ALTER TABLE `tblTowns`
  ADD PRIMARY KEY (`pmkTownId`);

--
-- Indexes for table `tblVotes`
--
ALTER TABLE `tblVotes`
  ADD PRIMARY KEY (`fnkNetId`,`fnkActivityId`);

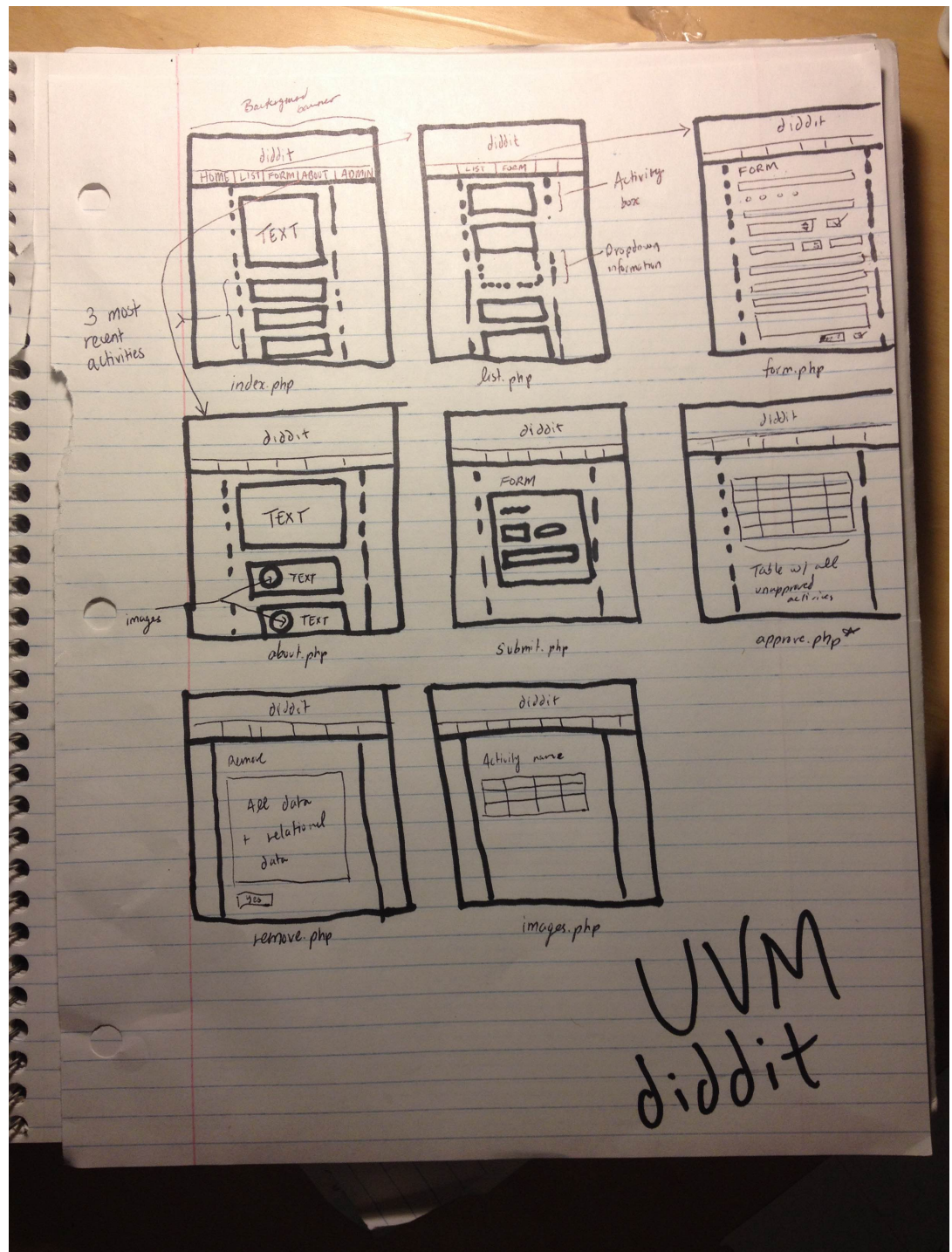
--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `tblActivities`
--
ALTER TABLE `tblActivities`
MODIFY `pmkActivityId` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=57;
--
-- AUTO_INCREMENT for table `tblPhotos`
--
ALTER TABLE `tblPhotos`
MODIFY `pmkPhotoId` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=25;
--
-- AUTO_INCREMENT for table `tblTowns`
--
ALTER TABLE `tblTowns`
MODIFY `pmkTownId` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=9;

```

Title of project	Version <1.0>
	December 12, 2015

## Story Board



Title of project	Version <1.0>
	December 12, 2015

## Specific requirements

The site will feature public pages, as well as admin-only pages, and we will separate the two categories of pages to make it easier to distinguish the public “face” of the site from the more “under-the-hood” features of the site when accessed by administrators.

Before we get into page-specific requirements, though, we will list several general requirements that apply to the entire site.

### General

1. The data created for and generated by this website will be saved in a database hosted by the University of Vermont. The database will be called “JSIEBERT\_UVM\_ACTIVITIES,” and it will feature six tables:
  - a. Affiliates
  - b. Administrators
  - c. Activities
  - d. Votes
  - e. Photos
  - f. Towns
2. In lieu of a logo image, we will include a banner (banner1.jpg) on every page that will be incorporated as a background image for the <header> element.
3. The navigation bar will appear over the banner image, and it will feature links to all pages on the site (excluding submit.php, which is better accessed from a specific activity link on list.php). The nav bar will feature a background effect on the link to the current page, as well as links that are hovered-over by the user.
4. The main content of each page will be encapsulated in a centered column that takes up two-thirds of the window (at desktop size).
5. The website should scale down to mobile devices. It will not necessarily be optimized for mobile devices, but it will rearrange to accommodate the smaller screen size.
6. The footer will identify the site’s authors and contain a link to the About page.
7. The site will use a CSS framework (Foundation) for the bulk of its styling.
8. The site will be maintained on GitHub, in an open-source repository: <https://github.com/bigSiebs/assignment10>.
9. Meta description for all pages: “This site displays present and past student's favorite activities from when they were students at the University of Vermont. It's meant as a loose road map for current students and a source of nostalgia for alumni.”

Title of project	Version <1.0>
	December 12, 2015

10. Meta character set for all pages: UTF-8

## Public pages

### index.php

1. Title: diddit | University of Vermont
2. The main page will contain text describing the site and its purpose.
3. Below this text, it will have a section called, "Recently Submitted Activities," which will call attention to the three mostly recently submitted (approved) activities. The list will not display images that have not been approved yet.
4. The page will contain a link to the full, rank list (list.php).

### list.php

1. Title: diddit | University of Vermont
2. The content of this page will be a list of activities submitted to our site, order according to the up and down votes they've received.
3. Each activity will be shown in a separate box, containing its ranks, buttons to up or down vote it, and its name, which will be linked.
4. When the user clicks the name link, the activity's box will expand, showing additional information about the activity. In addition, if any pictures related to that activity have been uploaded by user(s), then thumbnails of these images will appear in the box, and they user will be able to see the full version of the image by clicking on it. The box will contain a link that the user can follow to upload his/her own image pertaining to the activity.
5. The vote system will allow users to vote on activities, but it will restrict them to one vote per activity. They can change their vote, but the vote can ever have three values: 0, 1, or -1.
6. If voters are not in the Affiliates table, they will be added.
7. The list will show ten activities at a time.
8. **(Admin-only)** There will be links to edit/remove activities.
9. **(Admin-only)** Beneath the table, there will be a link for administrators to go to the approve.php page and approve activities.

### form.php

1. Title: diddit | University of Vermont
2. This page will contain a form for users to add activities to our database. The form will be the only content on the page.
3. The form will insert data into Activities, Towns, and Affiliates tables.
4. The form will contain the following fields and features:
  - a. a textbox to collect the user's NetID (this will be auto-generated and uneditable to non-admins)
  - b. four radio buttons for the user to specify their relation to UVM
  - c. a textbox for the activity's name



Title of project	Version <1.0>
	December 12, 2015

- d. a dropdown listbox for the category of the activity
  - e. a checkbox to indicate if the activity is on campus
  - f. a textbox for the town name
  - g. a listbox for the state
  - h. a textbox for the distance from Burlington
  - i. a textbox for a more detailed location description
  - j. a textbox for the cost of the activity
  - k. a textbox for the URL of the activity
  - l. a text area for a description of the activity
  - m. **(Admin-only)** a checkbox to approve the activity
  - n. a submit button for the user to send the information
  - o. validation of data and display of appropriate error messages
  - p. send email to user and admins
5. **(Admin-only)** The admins will be able to update records using the same form. An activity will need to be selected on list.php. An edit will not send an email to the user or the admins.

### **submit.php**

- 1. Title: diddit | University of Vermont
- 2. This page will allow users to upload images for activities that are listed in list.php.
- 3. Files will be stored in the uploads folder, and data will be added to the Photos table.
- 4. It will not be listed in the nav bar because it will be easier if users follow a link from list.php.
- 5. If an activity has been selected, the page will display the name of the activity and a form for an image upload. The form will contain several fields and features:
  - a. an upload box for users to submit a file
  - b. a text field for an image caption
  - c. a submit button to send the file and caption
  - d. validation of image file (to ensure its an image) and caption text
- 6. If an activity has not been selected, then the page will display a list of activities, from which the user can select and view the form for that activity.

### **about.php**

- 1. Title: diddit | University of Vermont
- 2. This page will be mostly information. It will describe the project, as well as the site's developers.
- 3. It will contain two images, one of each developer (alan.png and joe.png).
- 4. It will also contain contact information for the developers, should anyone want to contact them.

Title of project	Version <1.0>
	December 12, 2015

## **Admin pages**

The following pages cannot be accessed by non-admin users.

### **approve.php**

1. Title: diddit | University of Vermont
2. This page will display a brief overview of new, unapproved activities.
3. The records will be displayed in a table, with links to approve it, view/edit it (which will link to form.php, populated with the data), and remove it.

### **remove.php**

1. Title: diddit | University of Vermont
2. If an activity is selected, this page will display all the activity data, as well as relational data (any votes and/or images), all of which will be removed when the user clicks "Remove."
3. There will be a "Remove" button on this page, and the page will prompt the user only to click it if they're sure they'd like to remove all data.
4. If an activity is not selected, the page will explain this to the user, list all the activities in the database, and prompt them to choose one for deletion.

### **images.php**

1. Title: diddit | University of Vermont
2. This page is for admins to manage user-submitted images. It is a combination of the approve and remove pages for images.
3. If an activity with an image(s) has been selected, the page will display all associated images.
4. For each record, there will be a link to view the image, a link to approve it if it's unapproved (and vice-versa), and a link to remove it.
5. If the user opts to remove the record, a small form will appear at the top of the page and ask the user to confirm.
6. If an activity has not been selected, the page will display a list of all unapproved images, and a linked list of all activities with images submitted.