

# Unstable Slope Management Program (USMP) System Architecture Document

*(Version 1.2)*

Mike P. Wittie  
Montana State University  
[mwittie@cs.montana.edu](mailto:mwittie@cs.montana.edu)

# Table of Contents

## [1. Introduction](#)

## [2. Architectural Overview](#)

### [2.1. Amazon Web Services \(AWS\) deployment](#)

### [2.2. Resources Used](#)

## [3. Database](#)

### [3.1. Slope Information Form Data](#)

### [3.2. Maintenance Form Data](#)

### [3.3. New Slope Event Data](#)

### [3.4. User Account Data](#)

## [4. Web Server](#)

### [4.1. Data Privacy](#)

### [4.2. Data Security](#)

## [5. Webpage](#)

### [5.1. Data Quality Assurance](#)

### [5.2. Data search and export](#)

### [5.3. Links to other databases](#)

## [6. Mobile App](#)

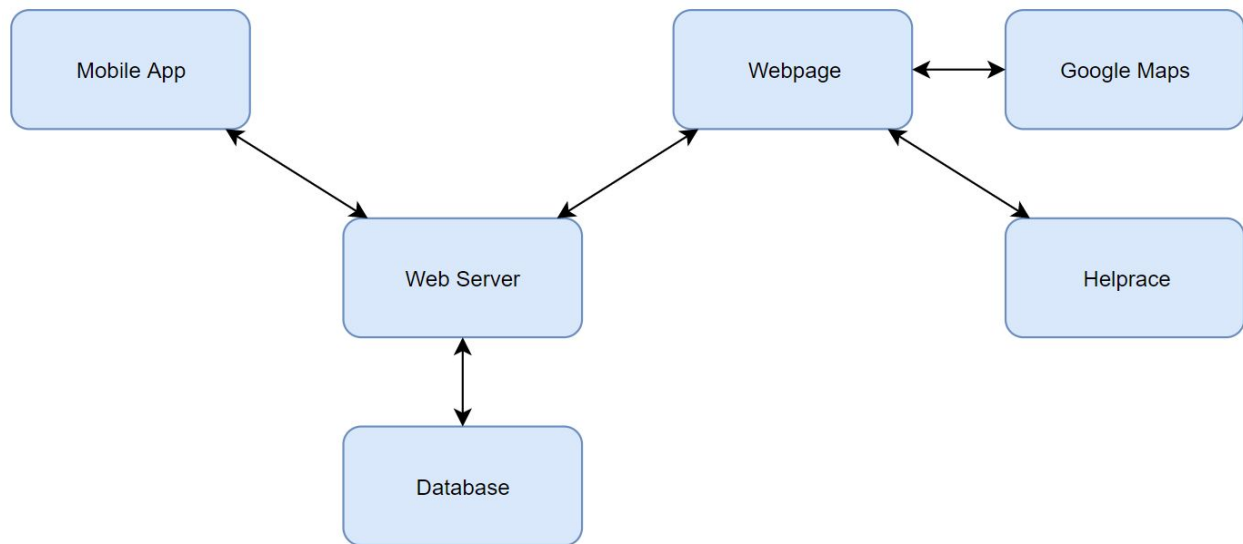
## [7. Support Ticket System \(Helprace\)](#)

# 1. Introduction

This document details the architecture and implementation of the USMP system. We describe the individual system components, including the database, server backend, website, and mobile app.

## 2. Architectural Overview

The architecture of the USMP systems contains a LAMP stack (a Linux server, Apache web server, MySQL database, and PHP programming language for server-side scripts), a mobile app built on React Native, and a webpage built on JavaScript/jQuery with a dependency on Google Maps. The website also connects with Helprace to support customer feedback and support tickets. The connectivity between the different components is shown below.



### 2.1. Amazon Web Services (AWS) deployment

Currently, the system is deployed on AWS. The web server functionality is divided between a production instance and a development instance. The production server servers pages on [usmp.info](http://usmp.info), while the development server instances on [testing.usmp.info](http://testing.usmp.info). The DNS entries for the system use [godaddy.com](http://godaddy.com) DNS, where the [usmp.info](http://usmp.info) domain is registered. The database runs on a separate, actively backed up, AWS RDS instance. There are two databases: 'usmp' supports [usmp.info](http://usmp.info), while 'usmp\_mwittie' supports the testing instance.

### 2.2. Resources Used

As of June 11th, 2019 the system uses the following resources:

- AWS EC2 compute nodes
  - two t2.micro instances
- AWS RDS database

- one db.t2.micro instance with 7MB of 20GB used
- AWS S3 document storage
  - images 5822 Objects - 5.9 GB
  - documents 4 Objects - 1.8 MB

The AWS hosting hosts have not yet exceeded \$40 in a given month, but could do so if the traffic to the site increases as the system gains popularity.

Additional required system resources are:

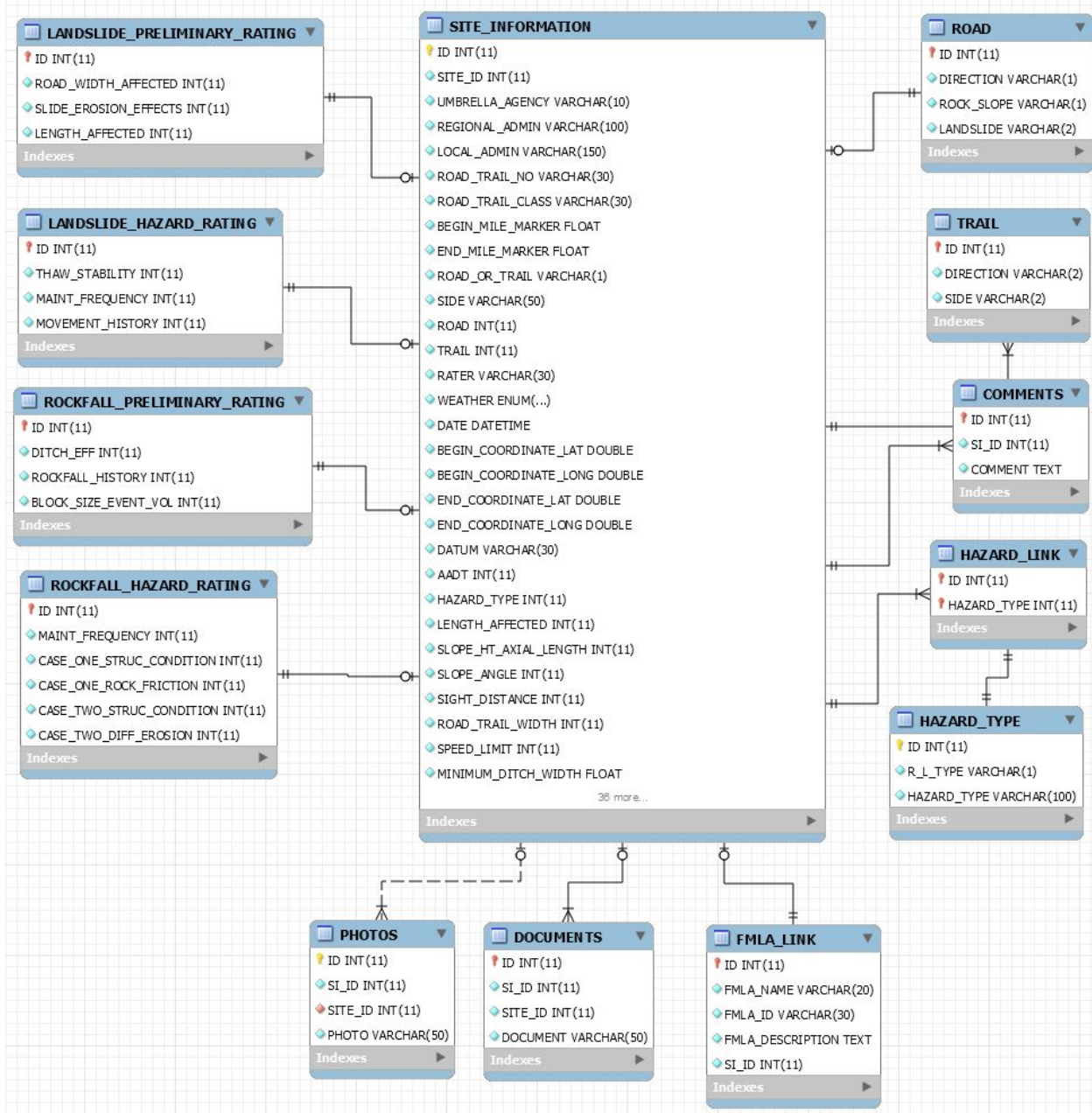
- iOS development license - \$99 per year
- Android development license - \$15 per year
- The system uses the free tiers/features from Google Maps and Helprace

### 3. Database

USMP relies on a MySQL relational database for storage of structured data recorded through the website and mobile app forms.

#### 3.1. Slope Information Form Data

Slope information data is stored in SITE\_INFORMATION table and several associated tables, as shown below.



There are several important constraints not represented in the data structure:

- A site (rated slope) has a unique `SITE_INFORMATION.SITE_ID`. Each edit to the site shares the same `SITE_INFORMATION.SITE_ID`, but has a unique `SITE_INFORMATION.ID`.
- The associated tables have foreign keys to individual edits onto `SITE_INFORMATION.ID`
- Each site has one or more `HAZARD_TYPE` values. The `HAZARD_LINK` table reflects this many to many relationship through its foreign keys.

- Each slope is either a Rockfall, or a Landslide. There is not field representing this in SITE\_INFORMATION. Instead a slope will have either an associated row in ROCKFALL\_PRELIMINARY\_RATING, or LANDSLIDE\_PRELIMINARY\_RATING.
- Uploaded images and documents are stored on disk, with file system paths to the images stored in the PHOTOS and DOCUMENTS tables.

### 3.2. Maintenance Form Data

Maintenance Form data is stored in the MAINTENANCE\_FORM table shown below.

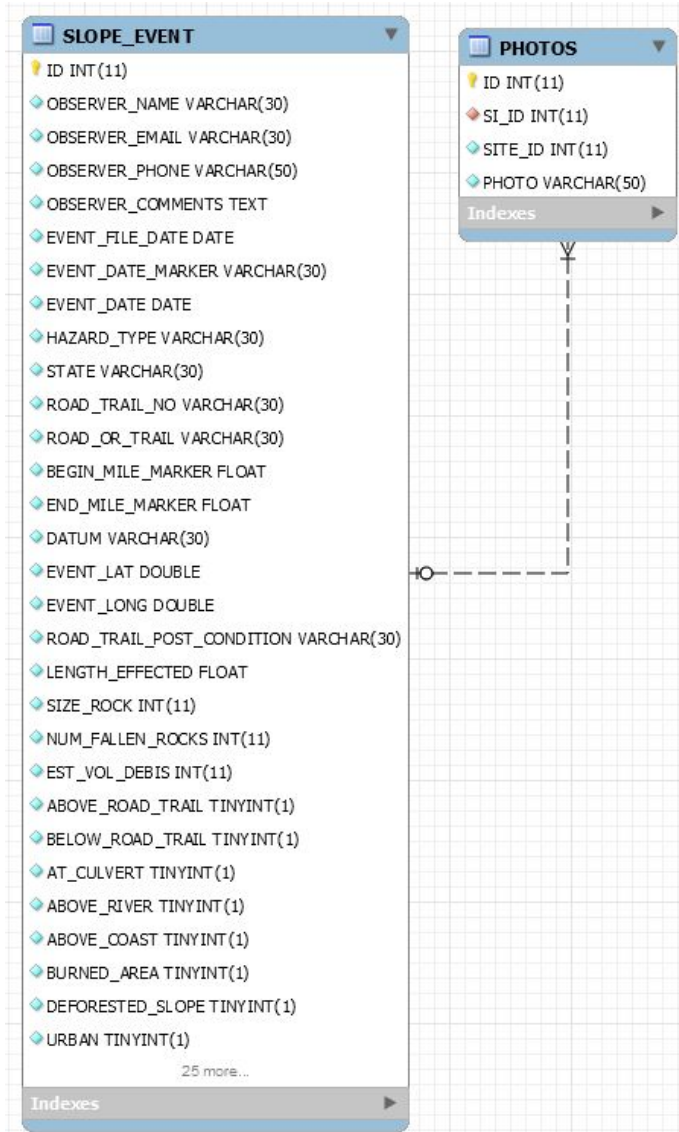
Field Name	Data Type
ID	INT(11)
MF_ID	INT(11)
SITE_ID	INT(11)
CODE_RELATION	VARCHAR(30)
MAINTENANCE_TYPE	VARCHAR(10)
ROAD_TRAIL_NO	VARCHAR(30)
BEGIN_MILE_MARKER	FLOAT
END_MILE_MARKER	FLOAT
UMBRELLA_AGENCY	VARCHAR(10)
REGIONAL_ADMIN	VARCHAR(100)
LOCAL_ADMIN	VARCHAR(150)
US_EVENT	VARCHAR(10)
EVENT_DESC	TEXT
MAINTENANCE_DATE	DATETIME
MAINTENANCE_LAT	DOUBLE
MAINTENANCE_LONG	DOUBLE
DESIGN_PSE	FLOAT
REMOVE_DITCH	FLOAT
REMOVE_ROAD_TRAIL	FLOAT
RELEVEL_AGGREGATE	FLOAT
RELEVEL_PATCH	FLOAT
DRAINAGE_IMPROVEMENT	FLOAT
DEEP_PATCH	FLOAT
HAUL_DEBRIS	FLOAT
SCALING_ROCK_SLOPES	FLOAT
ROAD_TRAIL_ALIGNMENT	FLOAT
REPAIR_ROCKFALL_BARRIER	FLOAT
REPAIR_ROCKFALL_NETTING	FLOAT
SEALING_CRACKS	FLOAT
GUARDRAIL	FLOAT

35 more...

Indexes

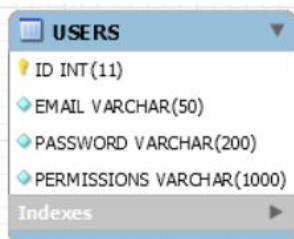
### 3.3. New Slope Event Data

Maintenance Form data is stored in the SLOPE\_EVENT table shown below. Photos for Slope Events are also stored in the PHOTOS table with SI\_ID as the foreign key.



### 3.4. User Account Data

User account data is stored in the USERS table shown below.



A user's EMAIL serves as their login. We store the hash of the password in USERS.PASSWORD, but not the password itself. USERS.PERMISSIONS stores the administrative domains, within which a user may update slopes. For example, a user whose

PERMISSIONS field contains 'NPS', may create and update forms for slopes located in areas administered by the National Park Service agency. The permissions may be more fine-grained. For example, a user may be restricted to the Rocky Mountain Region of the Forest Service with the PERMISSIONS set to 'FSRockyMountainRegion', or a single park, with the PERMISSIONS set to 'NPSIMRCHIRICAHUA'. Additional options include restrictions to States, Territories, and Counties.

The PERMISSIONS field may contain a combination of areas for users permitted to operate across them. This is particularly useful for users working in geographically adjacent administrative domains. The more general permissions, for example at the Agency level, take precedence over more restrictive areas, such as regions. And so, a user with permissions to operate in NPS, will not be restricted further if their PERMISSIONS field also includes the 'IMR' region.

Finally, the PERMISSIONS field may contain the keyword 'root' to give a user administrative privileges.

Users with administrative permissions may change the permissions of other users through the

A green rectangular button with the word "Account" in white text.

tab of the website.

To add a new user, the administrator enters their username and password and selects the permissions for the user in the **Permissions:** field. The 'Permissions' field is a multi-select field, where multiple permissions may be select with a control-click. To finish adding a user, the

administrator selects the  button.



## Add a New User

New Username:

New User Password:

Permissions: Press (ctrl+click) to select more than one

- Root
- BLM
- FS
- NPS
- OTHER
- Alaska Region
- Eastern Region
- Intermountain Region
- Northern Region
- Pacific Northwest Region
- Pacific Southwest Region
- Rocky Mountain Region
- Southern Region
- Southwestern Region
- AKR
- IMR
- MWR
- NCR
- NER
- PWR
- SER
- Alaska
- Arizona

Add User

To change the permissions of an existing user, the administrator selects the [Show users](#)

**Account**

link on the

Change

presses the

Update User

selection, the administrator presses the

button (which replaces the 'Add User' button) below the 'Permissions' field.

Finally, the administrator may delete a user by pressing the 'Show users' link, then scrolling down to the right users, and ultimately pressing the [Delete](#) [user](#) link next to their name.


## 4. Web Server

The code on the web server acts as a REST API for the webpage and the mobile app to make changes to the shared database state. The code is organized as follows:

- ▼ server
  - ▼ account\_management\_php
    - > P add\_user.php
    - > P change\_password.php
    - > P change\_permission.php
    - > P remove\_user.php
    - > P show\_user.php
  - ▼ authentication
    - > P app\_login.php
    - > P auth\_functions.php
    - > P initial\_login.php
    - > P logout.php
  - ▼ edit\_maintenance\_form\_php
    - > P edit\_maintenance\_form.php
    - > P get\_current\_maintenance\_form\_information.php
  - ▼ edit\_site\_php
    - > P edit\_site.php
    - > P get\_current\_site\_information.php
    - > P get\_documents.php
    - > P get\_permissions.php
    - > P get\_photos.php
  - ▼ edit\_slope\_event\_php
    - > P edit\_slope\_event.php
    - > P get\_current\_slope\_event\_information.php
    - > P get\_photos\_slope.php
  - ▼ maintenance
    - > P add\_maintenance.php
  - ▼ map\_php
    - > P get\_percentiles.php
    - > P get\_sites\_for\_caching.php
    - > P get\_sites.php
  - ▼ new\_site\_php
    - > P add\_new\_site.php
  - ▼ new\_slope\_event
    - > P add\_new\_slope\_event.php
  - ▼ search\_php
    - > P export\_search.php
    - > P search\_map.php
  - ▼ shared
    - > P delete\_photos\_docs.php
    - > P delete\_sites.php
    - > P get\_dropdown\_options.php
    - > P get\_hazard\_type\_dropdown\_options.php
  - ▼ site\_photos\_php
    - > P get\_photos.php
    - > P constants.php
    - > P image\_upload\_test.php
    - > P new\_slope\_form.php
  - index.html

These PHP scripts support API calls from the website and the mobile or provide supporting functions for these calls.

- Scripts in account\_management\_php support user management functions on [http://usmp.info/client/account\\_management.php](http://usmp.info/client/account_management.php)

- Scripts in authentication support user login and logout functions on <http://usmp.info/client/login.php>
- Scripts in new\_site\_php enable the creation of new slopes on [http://usmp.info/client/new\\_site.php](http://usmp.info/client/new_site.php). Scripts in edit\_site\_php, on the other hand, support the loading of information into the form at [http://usmp.info/client/edit\\_site.php](http://usmp.info/client/edit_site.php). These scripts also support fetching of data from tables associated with the SITE\_INFORMATION table.
- Scripts in new\_slope\_event\_php enable the creation of new slope events on [http://usmp.info/client/new\\_observer\\_form.php](http://usmp.info/client/new_observer_form.php). Scripts in edit\_slope\_event\_php, on the other hand, support the loading of information into the form at [http://usmp.info/client/edit\\_slope\\_event.php](http://usmp.info/client/edit_slope_event.php).
- Scripts in maintenance enable the creation of new maintenance forms on [http://usmp.info/client/new\\_maintenance\\_form.php](http://usmp.info/client/new_maintenance_form.php). Scripts in edit\_maintenance\_form\_php, on the other hand, support the loading of information into the form at [http://usmp.info/client/edit\\_maintenance\\_form.php](http://usmp.info/client/edit_maintenance_form.php).
- Scripts in map\_php fetch sites to put pins on <http://usmp.info/client/map.php> and provide information to select the correct icon color.
- Scripts in search\_php support map search functionality, but running queries based on criteria specified on the website, and with export functionality, by creating an export excel file.
- Scripts in shared support functions for deleting sites and photos as well as obtaining dropdown options for search functions. These scripts are experimental at this point.
- Scripts in site\_photos\_php support fetching photos from the database and the file system.
- constants.php holds website configuration, such as database paths and passwords.
- Image\_upload\_test.php is a demo program showing how to downscale images before uploading them to the server.
- The >  map\_thumbnails directory contains all the photos for rates slopes linked to from the PHOTOS and DOCUMENTS tables.

## 4.1. Data Privacy

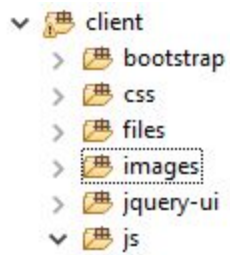
Personally identifying information (PII) in this system includes email addresses. These are associated with user login and are accessible to any system administrator with root level permissions. Additionally, the new slope event form collects observer name and email information. These data are available to any authorized user of the system.

## 4.2. Data Security

Login and session information is protected by the transport security layer (TLS) and asymmetric key encryption based on certificates configured and installed by <https://letsencrypt.org/>. Access to the database is controlled by AWS security groups firewall, which restricts raw access to the database only to developer machines and the web servers placed in specific IP subnets.

## 5. Webpage




























































The client code controls the functionality of the website.




The directories support the following functionalities:

- The CSS directory contains cascading style sheet files that control aspects of the website such as fonts, colors, and cell spacings
- The files directory contains files that are linked to from the website, including manuals and videos.
- Images contains all the image artworks for the website, including organization logos, icons, etc.
- The bootstrap and jquery-ui directories are necessary imports for JavaScript calls in the browser.

























The js directory contains most of the scripts that control the front end functionality. The directory is organized as follows:

- ▼  js
  - ▼  account\_management
    - >  account\_management.js
  - ▼  edit\_maintenance\_form
    - >  form\_submission.js
    - >  get\_add\_maintenance\_form\_information.js
    - >  get\_edit\_maintenance\_form\_information.js
  - ▼  edit\_site
    - >  get\_current\_site\_information.js
  - ▼  edit\_slope\_event
    - >  form\_submission.js
    - >  get\_add\_slope\_event\_information.js
    - >  get\_current\_slope\_event\_information.js
  - ▼  jquery
    - >  jquery-2.1.3.min.js
    - >  jquery-ui-timepicker-addon.js
  - ▼  localforage
    - >  localforage.js
  - ▼  map
    - >  get\_map\_sites.js
    - >  map\_search.js
    - >  set\_local\_id\_variable.js
  - ▼  new\_maintenance\_form
    - >  form\_submission.js
    - >  get\_new\_maintenance\_form\_information.js
  - ▼  new\_observer\_form
    - >  form\_submission.js
  - ▼  new\_site
    - >  get\_add\_slope\_rating\_information.js
    - >  show\_fields.js
    - >  show\_weather\_options.js
  - ▼  qra
    - >  chart.js
    - >  equations.js
    - >  field\_regex.js
  - ▼  search
    - >  search\_query.js
    - >  search\_text\_show\_hide.js
  - ▼  shared
    - >  agency\_info.js
    - >  delete\_photos\_docs.js
    - >  equations.js
    - >  field\_regex.js
    - >  form\_submission.js
    - >  get\_dropdown\_options.js
    - >  image\_upload\_test.js
    - >  ImageTools.js
    - >  include\_form\_table.js
    - >  include\_maintenance\_form\_table.js
    - >  include\_observer\_form\_table.js
    - >  sha1.js
    - >  url\_query\_string.js
    - >  validate\_form.js
    - >  validate\_maintenance\_form.js
    - >  validate\_new\_slope\_event.js
    - >  validate\_search.js
  - ▼  site\_photos
    - >  get\_site\_photos.js
  - >  open\_modal.js

The directories mirror the functionality implemented on the server. Scripts in each directory issue POST calls to the server .php files and then process the responses. There are a few important points to note:

- The js/localforage directory was used for offline storage on the website in its early version. This code is now disabled.
- The QRA folder implements the QRA module functionality completely on the client side.
- The shared directory has a number of scripts that supports functionality across the different forms. For example:
  - agency\_info.js and get\_dropdown\_options.js fetch a tree of agency affiliations for drop-down menus
  - equations.js implements calculations in form fields
  - field\_regex.js (along with field\_regex.file in individual folders) supports checking field formats and providing messages on how to correct them
  - ImageTools.js supports scaling down images before uploading to the server
  - include\* scripts enable inclusion of HTML forms, for example,  form\_table.html , in PHP scripts that show the different forms
  - sha1.js is used to rename files before upload
  - validate\* scripts enable input error checking on server side before upload

Finally, the website functionality is offered through a number of pages in the client/ directory:

- >  account\_management.php
- >  add\_maintenance\_form.php
- >  add\_observer\_form.php
- >  credits.php
- >  deny.php
- >  edit\_maintenance\_form.php
- >  edit\_site.php
- >  edit\_slope\_event.php
-  form\_table.html
-  image\_upload\_test.html
- >  login.php
-  logos.html
- >  maintenance\_form.php
- >  map.php
-  navbar.html
- >  new\_maintenance\_form.php
- >  new\_observer\_form.php
- >  new\_site.php
-  observer\_form.html
-  qra\_graph.png
-  qra\_table.html
- >  qra.php
- >  site\_photos.php
- >  welcome\_online.php

The \*.php files correspond to pages viewable by users. The \*.html files are the markup text for display and they are included by relevant \*.js files. For example, form\_table.html is included by both client/new\_site.php and client/edit\_site.php to display the slope rating form. The \*.html files

also call JavaScript/jQuery functions to trigger calculations and error checking whenever the input into the form changes.

## 5.1. Data Quality Assurance

The system implements data quality control (QC) functions during data entry. Each form field accepts data in a particular format that reflects the semantic of a form. The individual field formats are described as regular expressions in `field_regex.js` files. These files also contain error messages displayed to the user if their input does not match the prescribed format.

The website first checks field formats dynamically during data entry whenever a user switches to a different field and displays a browser alert with the appropriate format error message. Field formats are also checked whenever data is submitted. This second mechanism will show errors for required fields that have been left blank. The list of required fields is included in `validate_(form_name).js` files.

## 5.2. Data search and export

The website supports exporting of form data as comma separated value (CSV) files. To access the search functionality users click on the form-specific dropdown on the map page.

The screenshot shows a search interface for 'Slope Rating Information Search'. It includes a 'Reset' button and a link 'Right click on the Map to Add a new form'. The search criteria are defined by two dropdown menus, both set to 'Select Site Information', separated by 'AND' labels. The search operator is set to 'Equal'. There is a 'Type search From' input field. At the bottom, there are buttons for 'Search on Map', 'Export as CSV', and a checked checkbox for 'Latest edit only'.

They will be given a series of dropdowns to form a conjunctive query based on form fields. In the example above, a user may search forms that match a particular form ID, which is one of the fields in the slope rating form. Users may also select other fields and ranges for values, for example, the Preliminary Rating between 100 and 200.

This screenshot shows the same search interface as above, but with a range query. The first dropdown menu is set to 'Preliminary Rating', the operator is 'Between', and the value range is '100' to '200'. The second dropdown menu remains 'Select Site Information'. The 'Search on Map', 'Export as CSV', and 'Latest edit only' (checked) buttons are also present.

To export the data a user clicks the 'Search on Map' button, which will submit the search to the `server/search_map.php` script, which will transform the criteria into a SQL query. The results of the query will contain the map view to the sites that match the criteria. A user may then click the 'Export as CSV' button, which will send a request to the `server/export_search.php` script



which will generate and download the file containing data for the sites visible in the map window.

### 5.3. Links to other databases

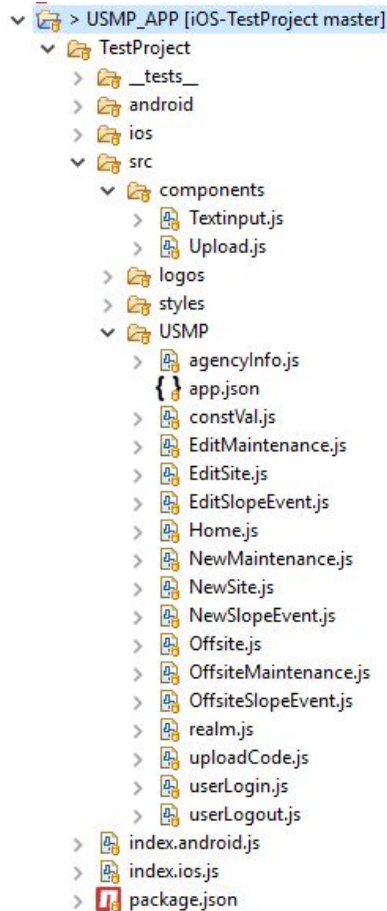
The slope rating form also supports links to with other database systems.

<b>Alternate database Name:</b> <input type="text"/>	<b>Alternate database ID:</b> <input type="text"/>
<b>Alternate database Description:</b> <input type="text"/>	

The above fields allow users to enter an alternate database name, form id, and description. These data may be used as keys to entries in other systems that describe the same slope and are visible in CSV files exported through the map search and reporting functionality.

## 6. Mobile App

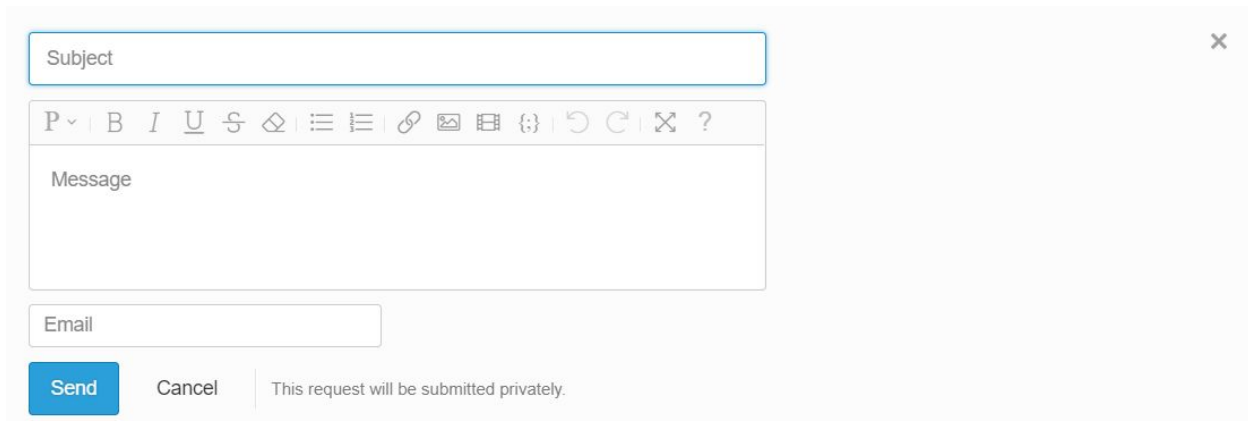
The USMP mobile application is implemented using the React Native platform, which compiles JavaScript code to create native Android and iOS applications. The code is organized as follows:



- The android and ios directories contain the compiled code for deployment through the Google Play and iOS App stores.
- The src directory contains the React Native implementation of the app
  - The components directory contains visual input elements used in forms for accepting input.
  - The logos directory contains agency logos used in the splash page of the app
  - The styles folder defines visual aspects of the application
  - The USMP directory contains the bulk of the app's functionality. The individual .js files correspond to the different app views (pages) For example, NewSite.js, contains both the form layout as well as calculations of the different fields. EditSite.js enables editing of a site that has already been created, whose data is pulled from the database by site id. Offsite.js allows loading of a site from for editing from local storage.  
Local storage is implemented using a Realm database, specified in realm.js. New forms stored in this database are submitted to the server using uploadCode.js. Finally, userLogin.js/userLogoug.js will implement login permissions for users. This functionality has been disabled for the current field trials.
- The index.android.js and index.ios.js are configuration files needed to generate iOS and Android app code for deployment.

## 7. Support Ticket System (Helprace)

To facilitate the interaction between USMP users and the development team, the website integrates with Helprace.com. Every page on the website has the [Feedback & Support](#) link in the bottom right corner. Clicking on the link brings up the following menu:



The screenshot shows a web-based form for submitting a support ticket. It features a 'Subject' text input field at the top. Below it is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), strikethrough (ABC), link (chain), unlink (chain with slash), list (bulleted), list (numbered), link (envelope), table (grid), code (code block), undo (curved arrow left), redo (curved arrow right), and a help/question mark icon. The main body of the form is a large text area labeled 'Message'. Below the message area is an 'Email' text input field. At the bottom left, there is a blue 'Send' button, a 'Cancel' button, and a note that reads 'This request will be submitted privately.' A close button (X) is located in the top right corner of the form's container.

To submit a question, suggestion, or an issue, the user can fill out the Subject and Message fields. Submissions may be made anonymously, or if the user expects a response, with the Email field filled out.

Helprace will notify the development via email. Our goal is to reply to all inquiries within 48 hours. If the inquiry requires changes to the system implementation, we will notify the user that their issue has been included in our internal tracking system. We will then notify them when we have a time estimate on a resolution once the issue makes it into our next development sprint. Finally, we will notify the user when the solution to their issue has been implemented and what steps they need to take, if any, to use the newest version of our software.