Individual Assignment Specifications

- Team: Memholes
- Iteration: Milestone 1

Special Roles

- Project Coordinator: Danny Chamberlin
- Quality Assurance Czar: Thomas Goodman
- Video Demo Creators:
  - Andy Ta, 999
  - Danny Chamberlin, 999
- Demo-Booth Operator: Andy Ta

Tasks:David Reddick

Task 1: Update Pothole

- Description: Make webpage to update reported pothole information.
- How to Evaluate: Able to edit pothole information from website and have database updated.
- Outcome of Task: User is able to edit pothole information.

Task 2: Name Pothole

- Description: Make webpage to update reported pothole name.
- How to Evaluate: Able to edit pothole name from website and have database updated.
- Outcome of Task: User is able to edit pothole name.

Task 3: Verify Pothole

- Description: Update reported pothole as verified.
- How to Evaluate: Able to mark pothole as verified from website and have database updated.
- Outcome of Task: User is able to mark pothole as verified.

Task 4: Duplicate Pothole Report

- Description: Update reported pothole as duplicate for later merger.
- How to Evaluate: Able to mark pothole as duplicate from website and have database updated.
- Outcome of Task: User is able to mark pothole as duplicate.

Tasks: "Ty" Thomas Goodman

Task 1: Embed Map into Home page

- Description: Make the map a part of the home page.
- How to Evaluate: If the map appears in the home page.
- Outcome of Task: Map appears on the home page.

Task 2: Retrieve location data from database

- Description: Query the database for pothole locations. This is needed to make pins display and to list locations.
- How to Evaluate: If database queries are used in the code and adding locations shows results in tasks 3 and 4 below.
- Outcome of Task: Database queries function and the location data reaches the map and list box.

Task 3: Pass location data to map API for display

- Description: Use a map API with the retrieved location data to display pins on the map.
- How to Evaluate: If the locations are visually pinned on the map.
- Outcome of Task: The locations are displayed as pins on the map.

Task 4: Pass location data and other details to a list box

- Description: List the retrieved location data in a box.
- How to Evaluate: If the box containing locations appears and lists items.
- Outcome of Task: The list box displays the pothole locations.

Tasks:Andy Ta

Task 1: Get Longitude and Latitude of selected location

- Description: Get the longitude and latitude of selected location.
- How to Evaluate: Longitude and Latitude is added to the database.
- Outcome of Task: User is able to click on any location on the map and have the latitude and longitude pre-populated into submit form.

Task 2: Update entry in database

- Description: Select and update rating entry in database.
- How to Evaluate: If the user is able to select a pothole and change its rating from 1 to 10.
- Outcome of Task: User is able to change the rating for potholes.

Task 3: Query the database for fixed pothole

- Description: Select fixed pothole ID in database
- How to Evaluate: Website should only display fixed pothole when select view fixed pothole.
- Outcome of Task: User is able to select the option for fixed pothole and website is displaying only fixed pothole.

Task 4: Query the database for verified pothole

- Description: Select verified pothole ID in database
- How to Evaluate: When user select this mode, the website should only display verified potholes.
- Outcome of Task: User is able to select option for verified pothole and website is displaying only verified pothole.

Tasks: Danny Chamberlin

Task 1: Create Login Page

- Description: Create View to allow user to log into application
- How to Evaluate: Login.html page is present.
- Outcome of Task: User can log in with a login page link.

Task 2: Wire login page to database

- Description: Login page will be wired to the database to allow login to site
- How to Evaluate: User logs in with previously established username and password and is successfully redirected to the home page with username showing on page.
- Outcome of Task: Users actually can use the log in page and get to their account.

Task 3: Create 'Create Account' Page

- Description: Create view to allow user to create a persistent account on site.
- How to Evaluate: AddUser.html page is present
- Outcome of Task: User can create an account.

Task 4: Wire 'Create Account' page to database

- Description: 'Create Account' page will be wired to the database to allow creation of user profile for site
- How to Evaluate: User fills out form with required information and is redirected to home page with "account created successfully" notification at top of page.
- Outcome of Task: Users can create accounts.

Task 5: Create query object for user's submission list

- Description: Create a query object in Pothole Info model to get list of potholes that belongs to currently logged-in user's user_id.
- How to Evaluate: PotHole Info model has above described query object.
- Outcome of Task: Users can see their submissions.

Task 6: Add link to user's potholes on profile page

- Description: Add a link to the user's profile page that will redirect to home page and show list of current user's potholes
- How to Evaluate: User clicks on link and list of potholes owned by that user_id is displayed on home page.
- Outcome of Task: User can view their potholes' show/edit pages.

Task 7: Pass Location data to map to display pins

- Description: Pass location data from database to the map object
- How to Evaluate: This and next task will be evaluated at the same time
- Outcome of Task: User's pins appear on the map.

Task 8: Display pothole data below map

- Description: Show pothole information in table below map

- How to Evaluate: User's list of potholes will be shown below map, while pins will be shown on map that correspond to each pothole in the table.
- Outcome of Task: User's pothole data appears below the map.

Task 9: Add Bootstrap to project

- Description: Add Bootstrap styling to application via process described in Ruby Bootcamp.
- How to Evaluate: Page CSS will have Bootstrap naming conventions, bootstrap gem will be present in project.
- Outcome of Task: Site has bootstrap styling.