# RiverNET Water Quality Documentation

## **Documentation Contents:**

[Class Diagram](#_jadljggx0c2r)

[Important Classes](#_457kqwlupqc2)

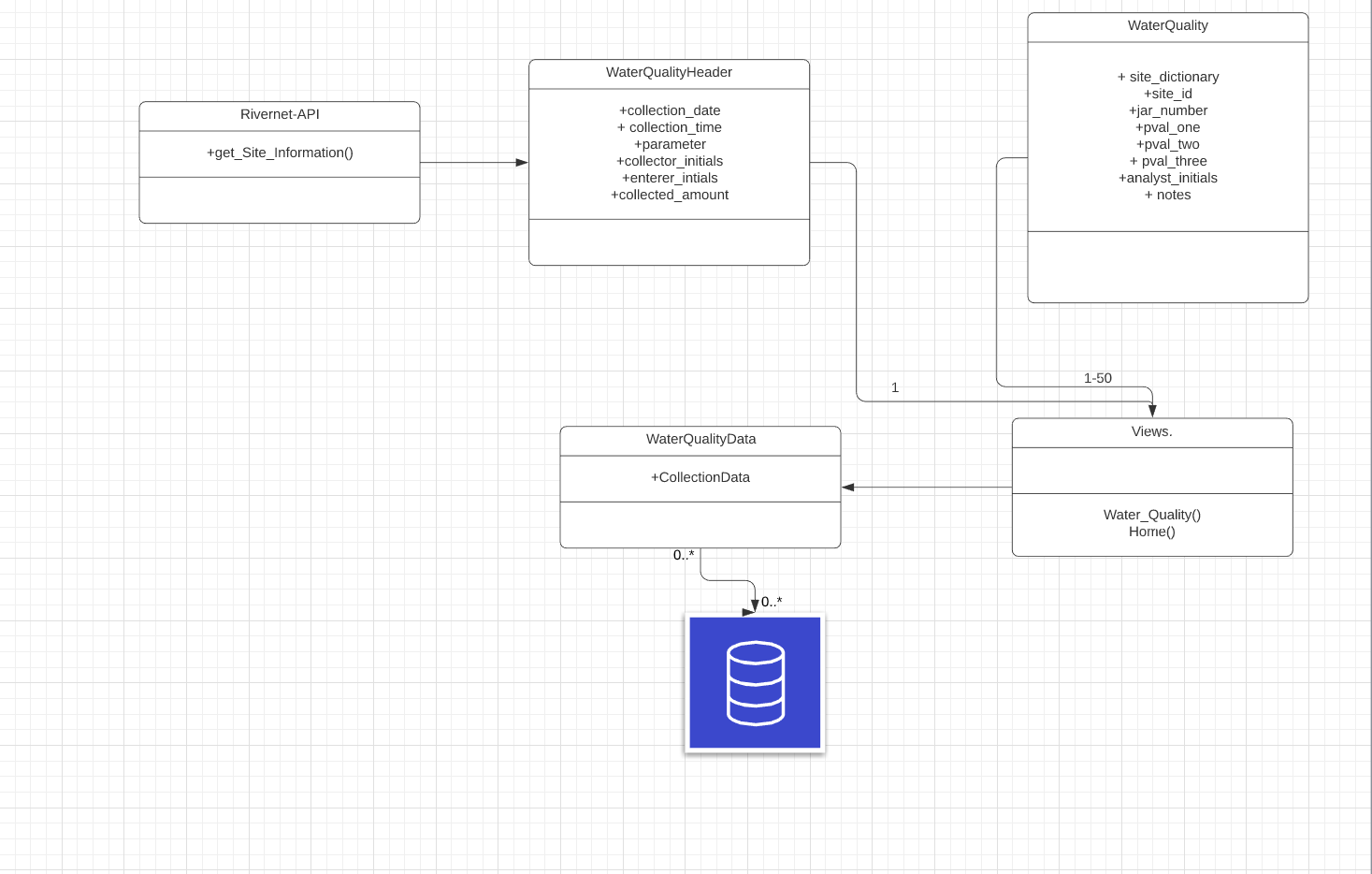
[Django and Folder Structure](#_a4f0d4ai835r)

[Testing Information](#_m9o4y7qwo13g)

### 

### 

### Class Diagram



### 

### 

### Important Classes

**Model, Forms, View**

Data collection in the Rivernet application is implemented via the files model.py, forms.py, and view.py. Each of which is located under the folder polls. In the file, model.py, the database structure is initiated as an object-relational database mapping via the WaterQuality object. Here the single line code means that each Rivernet row of data to be entered will be defined as a single JSON file. The forms.py file object, WaterQuality, defines each data attribute and its object type. From here, forms.py gets implemented by view.py and the HTML documents in the folder “Template.” Once the values are filled by the user, the ‘OK’ button at the bottom defines a POST method which callsviews.py calls a POST method which acting on the WaterQuality data members, creates a JSON dictionary and saves those values by passing them into an instance of model.py’s WaterQuality object .

### 

### 

### 

### 

### Django and Folder Structure

**File Structure**

-src

-mysite

(systems level configurations

-polls

(app level files)

-static

-main.py

-manage.py

-requirements.txt

The Django development framework works by compartmentalizing system wide files from site specific files. In the folder mysite, the files settings.py, urls.py, and wsgi.py, each define systems wide configurations, including all database, server, and root level build configurations. The file polls contains all of the application logic including,rendering capabilities, URL routing, object relational database mapping, and form collection. The folder static contains all CSS files, which get looked for by defining static folder root url in settings.py. Finally, main.py tells the application where and how to start the build process, while manage.py is a django specific command tool that can be used for a variety of administration tasks.

### Testing Information

There are various tools of testing used in Rivernet. Selenium Suit Testing, PyTest, and Travis CI.

Selenium is an automated testing framework used to validate web applications in different browsers. It creates an HTTP request for each selenium command and sends it to the browser driver. The Selenium tests for Rivernet are all run by the command line runner. The command line has the ability to pass different configuration arguments at run time.

Pre-requisites

-node

-npm

-selenium-side-runner

Pytest framework supports complex functional testing for applications and libraries. Pytest is used in Riverent to test Forms, Models, Database and views of the Django Framework. Pytest can run a couple of function tests. Basically the assertions check that returns true or false status. If an assertion fails in a test method then the execution is stopped. In this course we are required to use JUnit for testing But Django does not support JUNIT. Pytest is the tool is the equivalent to it.

Travis CI is a continuous integration service used to build to test projects hosted by GitHub. It automatically detects when commit has been made and pushed to a repository. We test the functionality of the updates made to the environment after every commit and push.