**Water Management Service**

Rachel Friedman | rfriedman113@gmail.com

CISC 3810 | Project 2 | Proposal

**Software Description**

The software I’m working on allows users to view water charges from 2013 until present. The data, provided by the New York City Housing Authority, is organized by borough and by development, then by building and it includes utility vendor and meter information, as well as all water and sewer charges from 2013. It offers user authentication and supports two different types of users; users from the NYCHA can log in and view data for all develoments, while owners of developments can log in and view only their own data. Theoretically, the second set of users will be able to use this software to either pay their bills or keep track of all their charges and payment status.

This type of software is interesting to me because the data can be broken up and organized in multiple different ways. NYCHA users will have the option of viewing a list of developments in a borough, or of viewing buildings in a development and the associated charges. Development users will be restricted to viewing only their buildings. Implementing this will allow me to experiment with SQL commands such as joins and views, and to incorporate Python’s SQLAlchemy as an Obejct Relational Model. Furthermore, since the information available to a user will be context dependent and varying depending on user type, this software will also make use of Python’s Flask-Login library to implement user authentication.

My project is similar to services offered by utility companies that are intended to allow clients to pay their bills or manage records. For example, see [Home | R.E.M. Residential | Property Management Company NYC (remny.com)](https://remny.com/) and [ACI Payments, Inc](https://www.officialpayments.com/pay-utility-bill-online.jsp) However, one thing that is unique about my software is that that the data is of a larger scale than the previously listed examples, with data going back to 2013. It seems likely to me that something similar for the whole state does exist but is perhaps not publicly available. Still, the implementation is similar and it’s an interesting exercise to imagine how one manipulates such a large dataset and designs an application with which to access the relevant data, while restricting access to other data.

**Application Diagrams and how it works**

**HOME PAGE**

A picture containing graphical user interface

Description automatically generated

**LOG IN**

Graphical user interface, application, Word

Description automatically generated

**DEVELOPMENT USER PAGE after logging in**

Developer is presented with list of managed buildings and can then click on View Charges to view charges associated with each building.

Table

Description automatically generated

**VIEW CHARGES (after clicking on one of the View Charges above)**

**Graphical user interface, application

Description automatically generated**

**PAY BILL (After clicking on Pay Bill in previous screen)**

**Graphical user interface, application

Description automatically generated**

**VIEW ALL DEVELOPMENTS (NYCHA Admin View)  
Admin users can click on a development to get a list of buildings, as in diagram #3.**

**Graphical user interface, table

Description automatically generated**

**Database Schema**

