CS4400 Database Project

Spring Semester 2017

Version 1.1

SLS (Service, Learn, Sustain)

Look at the last page for edits made for each version. Please read the entire description of the project before starting to work on it.

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an online system. You are required to use the classical methodology for database development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. ***In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database. You also cannot use any other software like Access.*** Ask professors or TAs if you have doubts in which tools/languages/software are allowed.

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources). Slides on database design methodology will be useful for phases I and II: All slides have been posted on T-square.

Regrade Policy

Once graded phases and/or quizzes are returned, there is a one-week deadline during which you can contest your grade. This clock starts not when you personally get your returned paper, but when the papers are returned to the class.

Groups

Project groups must have 4 members. You are allowed to form groups across the two sections (A & B) of the class. A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written notification with a proper justification must be provided to the professor and the head TA at that time in hard-copy form.

Deliverables

Phase I (submit on TSquare and bring a hard copy to class)

The deliverables include (put everything in one pdf file):

1. A cover page. Your cover page **MUST** include all information on the template. (Template is under Resources/Project)
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints (at least 3). You are required to include at least three logical constraints, although a fully-specified system will probably have more than that.
5. Any assumptions made with explanations.

Notes:

1. The EER must capture the functionalities of the system whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must have all functionalities. You are allowed to make up additional assumptions as long as they do not conflict with the specified constraints and requirements. You must list all your assumptions; otherwise TA would mark your ER diagram wrong since they would not be able to know you had made your own assumptions.
3. Logical constraints that can be specified directly using ER notation will not count towards the three required. Constraints related to data type or value are not accepted as constraints.

Each group needs to turn in one hard copy (only one for the entire group), and each group member should upload an electronic copy on T-Square individually. You will receive -5 penalty if you do not submit an electronic copy. **Please write down your Group Number clearly on cover page.** If you do not know your group number, please email the head TA.

Phase II (submit on TSquare and bring a hard copy to class)

The deliverables include (put everything in one pdf file):

1. A cover page same as phase1
2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution

provided)

3. Relational Schema Diagram (Identify primary and foreign keys and show referential integrity using arrows)
4. Create Table statements, including domain constraints, integrity constraints, primary keys, foreign keys and appropriate referential triggered action clause.

Note:

1. **Only one hard copy** should be turned in for the entire group, and each group member should upload an electronic copy on T-Square **individually**. **You will receive -5 penalty if you do not submit an electronic copy. Please write down your Group Number clearly on cover page.**

Phase III (Submit on TSquare)

The electronic deliverables include:

1. A cover page same as phase1 & 2.
2. A text file with all SQL statements for each task (follow the template in the phase II design methodology)
   Note: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks.
3. For heavy weight option, you also need to submit your source code.

Notes:

1. Prior to the demo, the TAs will give guidelines for populating the database with data. The database has to be populated with this data set prior to the demo.
2. **Each group member should upload an electronic copy on T-Square individually. You will receive -5 penalty if you do not submit an electronic copy.**

On demo day:

Bring your laptop and make sure you have a text file on your laptop with all your SQL queries just in case your application does not work. More details about demo will be discussed later this semester.

Grading

The project will consist of three phases (deliverables) as well as a final demo to the TA.

Phase I and Phase II of the project are each worth 10% of your final grade.

Phase III (20% for heavy-weight or 5% for light-weight, depending on option):

> **Heavy Weight Option (20 %):** The students would be required to use the embedded SQL feature of MySQL which allows you to embed SQL statements in a standalone application.
>
> **Light Weight option (5%):** The students would be required to demo the SQL queries on the MySQL console. Those who choose the light weight option would be required to take the Final exam.

Note that you can always change your option until the demo starts. Once TA starts to demo your project, you cannot change heavy-weight option to light-weight or vice versa.

Final Exam (15%): This would be only taken by students who have opted for the lightweight phase III. Under no circumstances would a heavy weight option student be allowed to take the Final.

Project

For this project, you will create a tool that stores places of interests which are related to SLS (Service, Learn, Sustain).

The following sections contain a functional description of the system along with some mockup screens. Each section would explain a functionality and then present an example screen about it. **You don't have to follow the UI designs, but your program needs to support all the functionalities.** These mockups are just for helping you to understand all the functionalities. A complete reorganization of the user interface is permissible as long as your application supports all the functionality listed below. The sections have been grouped by customer's functionalities and managers' functionalities.

For heavy option, you may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). We will also send an announcement about which languages/tools/software/platforms are allowed later this semester. (Do ask the professors for permission if in doubt.)

All Users Functionality

1. Log In

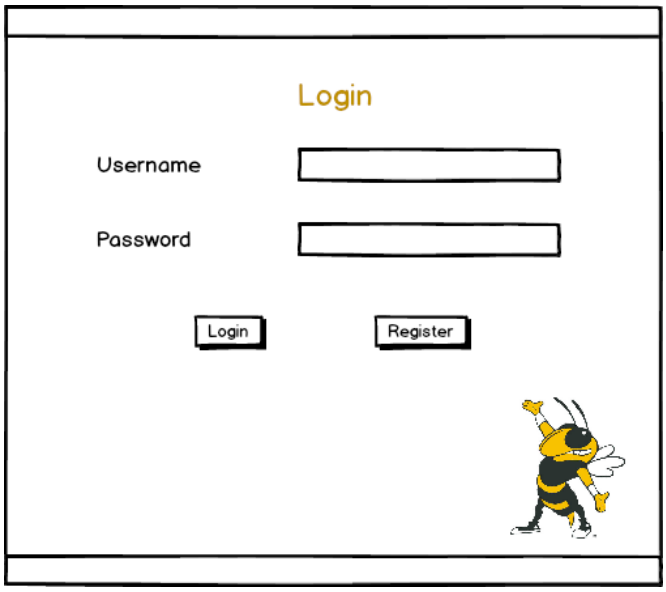

Fig 1: Log in

Fig 1 shows the login screen. All users must login before using this application. There are three types of users: city scientists, city officials and admins. To login, a valid username and password combination is required. If users provide invalid login credentials, an error message should be shown on the screen.
If users do not have account yet, they can click on the register button to create an account.

Notes:

1. *Username* is **unique** for every user.
2. Since all three types of users share the same login screen, you need to check if the user is a city scientist, city official or an admin.

2. New User Registration

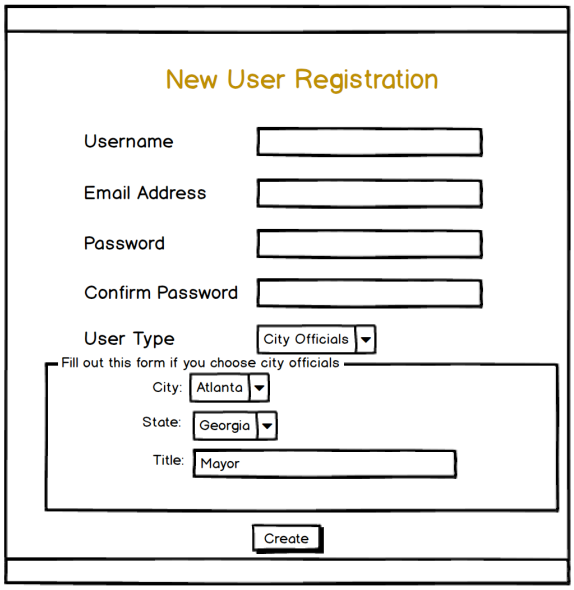

Fig 2: New User Registration

After clicking register button in Fig 1, users will be directed to this registration page. All fields must be filled.

Requirements:

- *Username* must be **unique**
- *Email* must be **unique**
- *Password* and *confirm password* must match
- *User Type:* choose one from *city officials* and *city scientists*. We assume all admin information have been stored in the database. (In other words, you cannot create admin accounts)
    - If users choose “city officials”, they also must enter *city, state* and *title.*
        - You can assume the combination of *city* and *state* is **unique**
        - *City* and *state* are pre-stored in the database
        - All city official accounts must be approved from admins. (We will talk about this later)

An error message should be displayed if any requirement is not met.

City Scientists Functionalities

1. Add a New Data Point

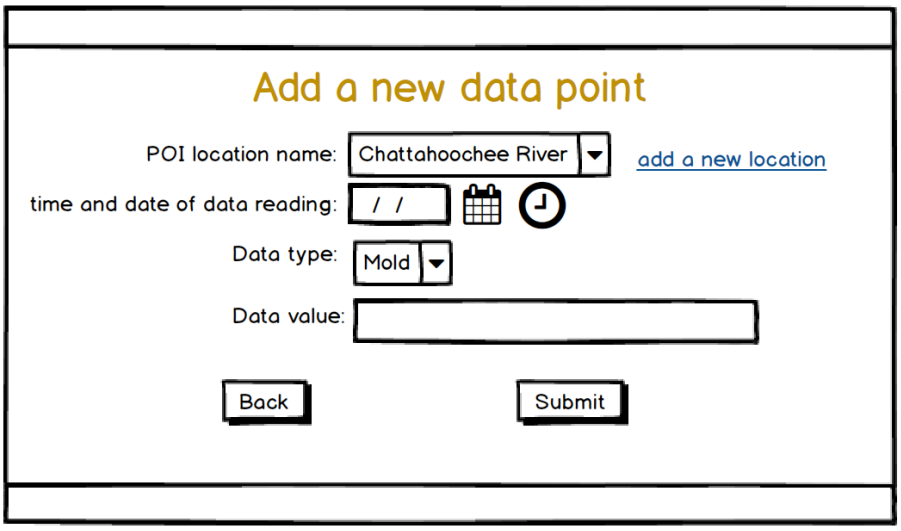

Fig 3: Add a new data point

City scientists can add a new data point to Point of Interest(POI). They can also create a new POI location. All fields must be filled.

Data:

- *POI location name*: (for example: Chattahoochee River, etc) All locations should be stored in the database.
- *Time and date of data reading*: (for example: 01/31/2017 15:34)
- *Data type*: Select either *Mold reading* or *Air Quality reading* (You should not hardcode values for dropdown menu. Both data types need to be stored in the database)
- *Data value:* (for example: 145. An integer)

2. Add a New POI Location

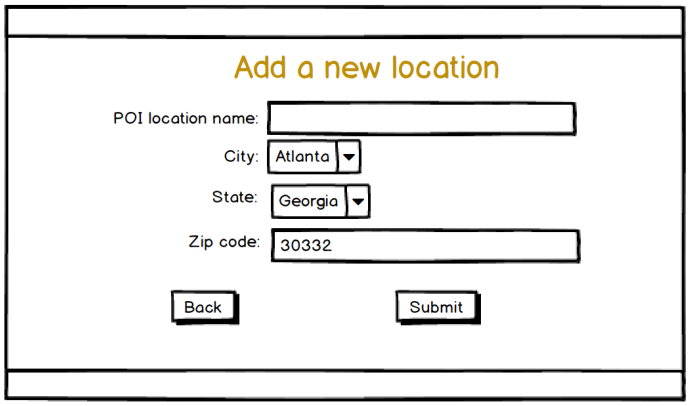

Fig 4: Add a new location

Data:

- *POI location name*: You can assume location name is **unique**.
- *City* and *State:* You can assume the combination of *city* and *state* is **unique**.
- *Zip code:* One POI location has only one zip code.

Admin Functionalities

1. Choose Functionality

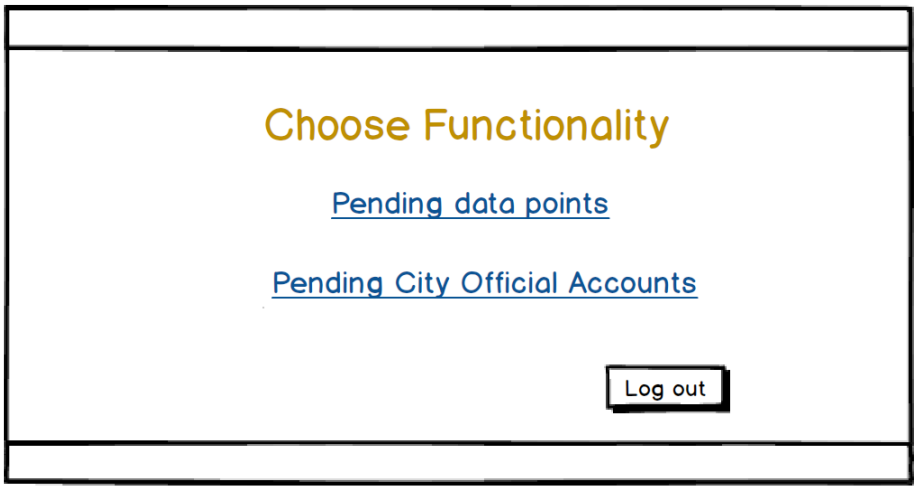

Fig 5: Choose Functionality (Admin)

2. Reject or Accept a Data Point

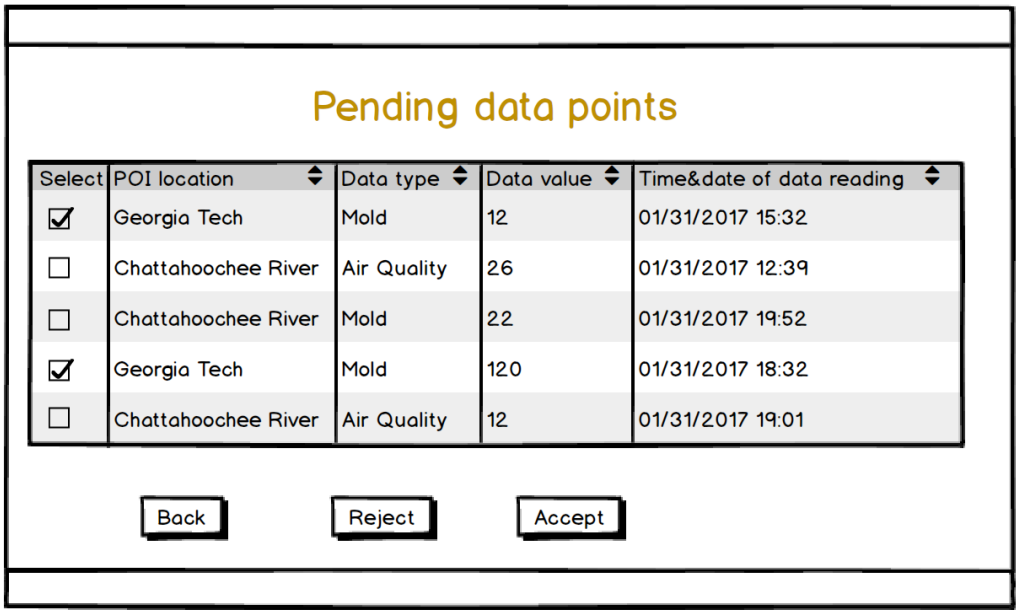

Fig 6. Pending POI data points.

Admin can view all pending data points, and reject/accept data points.

Admin can also sort pending data points by *POI location, data type, data value* and *time&date of data reading*

- You can assume for one POI location, each data point has a **unique** *time&date of data reading*

3. Reject or Accept City Official Accounts

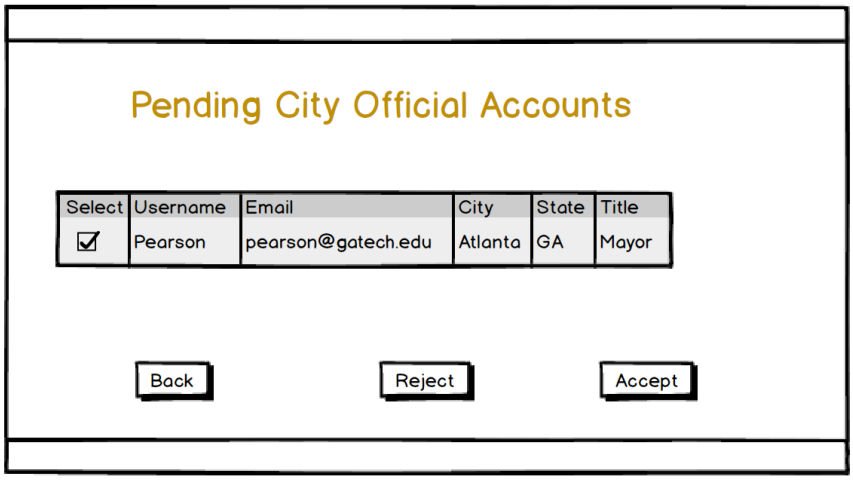

Fig 7. Pending City Official Accounts

Admin can view all pending city official accounts, and reject/accept them.

City Official Functionalities

1. Choose Functionality

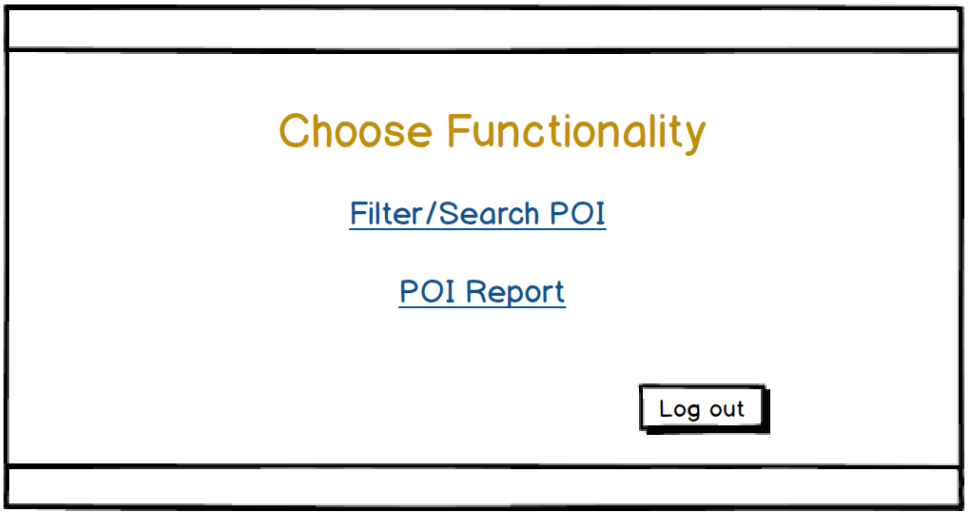

Fig 8: Choose Functionality (City Official)

2. Filter/Search POI

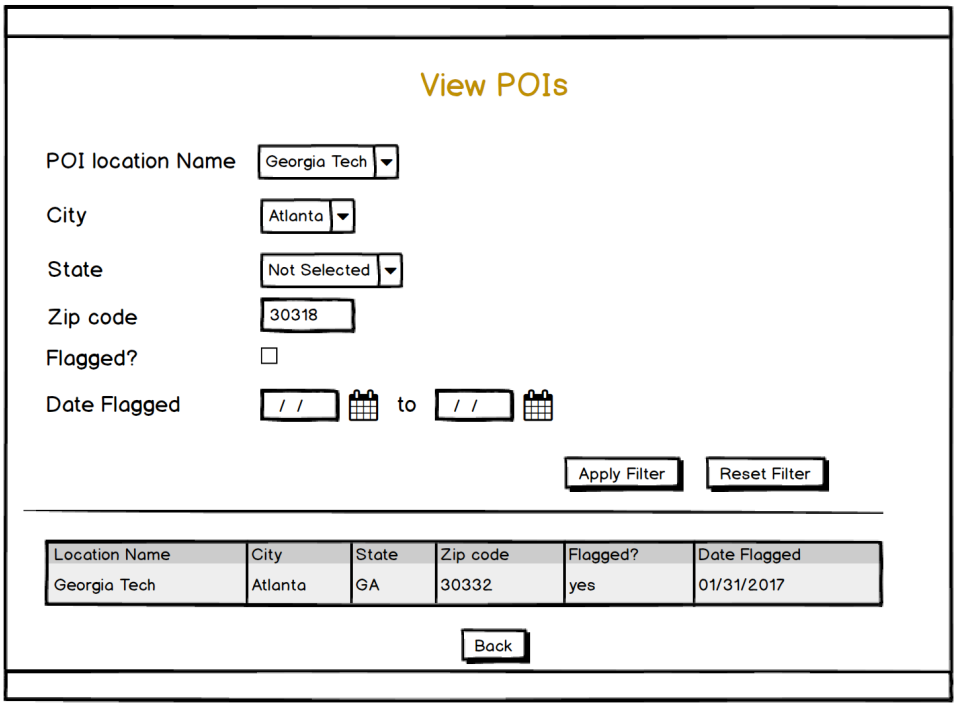

Fig 9. Search/browse POIs

City officials can browse and search for places of interest. They should be able to filter based on the criteria listed below. City officials should be allowed to select and view a read-only version of the POI from this list.

You must support the following filters, at a minimum:

- *Location Name*: Search by location name (Chattahoochee River, etc)
- *City:* Search by city
- *State:* Search by state
- *Flagged*: checkbox indicating whether to show only POIs flagged by city officials (will talk about flag later)
- *Date Flagged*: Show only POIs that were flagged between this data range

City officials can view POI detail by clicking on the result.

3. POI detail

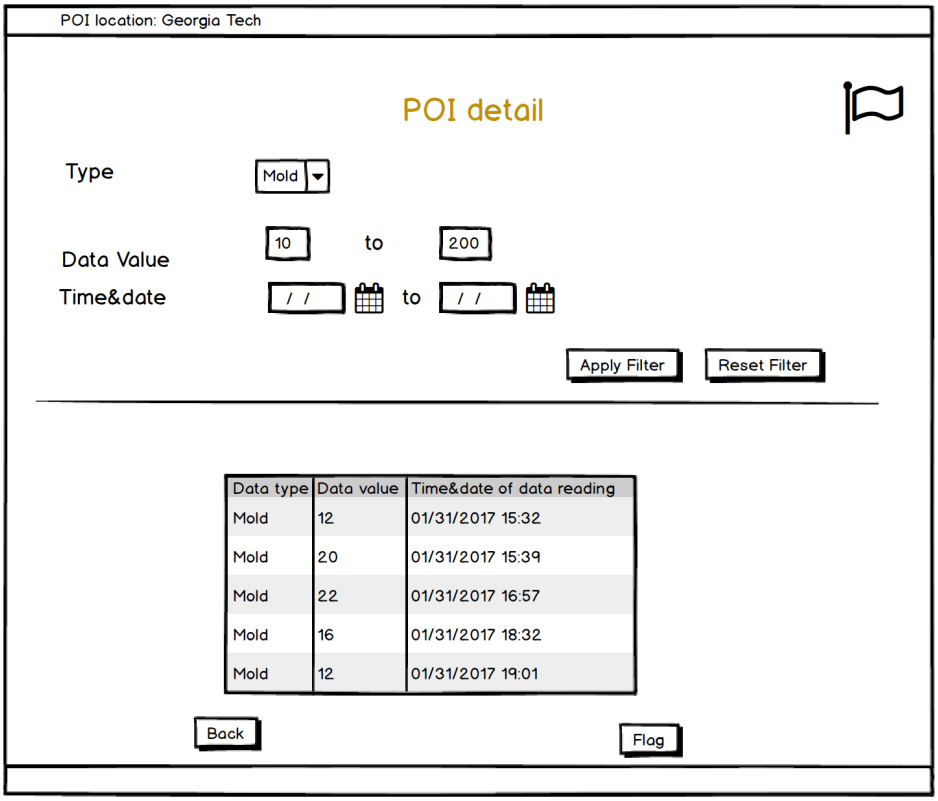

Fig 10. POI detail

A place of interest (POI) should show location, and a list of data readings sorted by *time&date of data reading*. A data reading has a type, a value, and the date & time that the reading was taken.

You must support the following filters, at a minimum:

Type: Mold or Air Quality

Data Value: Enter a range

Time&date of data reading: Show data points recorded between this time range

City officials can also add and remove a flag to the POI, indicating that they are looking into the issue. When the flag is set, the date it was flagged is also stored.

4. POI Report

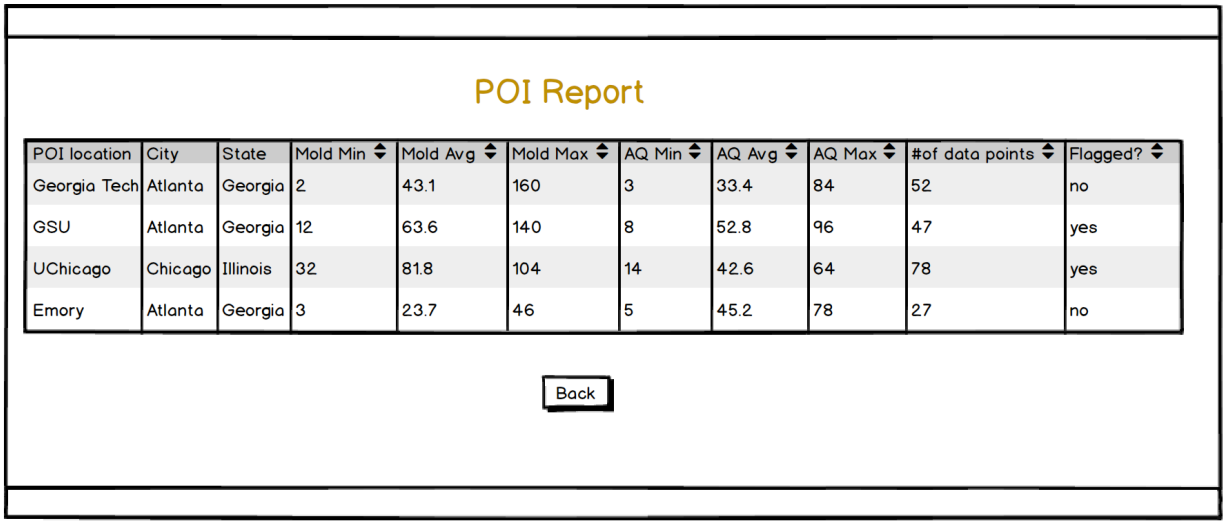

Fig 11. City Official Report

City officials can generate a report of all POIs. The POI report should include the minimum, average and maximum value of each type of data reading from that POI, number of data points and all other values shown in the mockup.

City officials can sort POIs by *Mold Min, Mold Avg, Mold Max, Air Quality(AQ) Min, AQ Avg, AQ Max, Number of Data Points* and *Flagged*?.

1.

Date	Version	Note
1/31	1.0	
3/7	1.1	Phase2: CREATE table statements should have appropriate referential triggered action clause