Milestone 3 Instructions

Dave Anderson, David Nguyen, Zhuojing Xie

Project 4

COVID-19 Management System

Connecting to the Database:

To run queries in the console or GUI application you need to have the mm_cpsc502102team04 database set up in your local connection. In your MYSQL workbench go to your local connection and open the milestone3dump.sql file. Run the SQL script and it will automatically create a new schema called mm_cpsc502102team04.

If you would like to instead use the database on the cloud you can access it through the Seattle U VPN and these credentials:

Hostname - cs100.seattleu.edu

Username - mm_cpsc502102team04

Password - mm_cpsc502102team04Pass-

Database: mm_cpsc502102team04

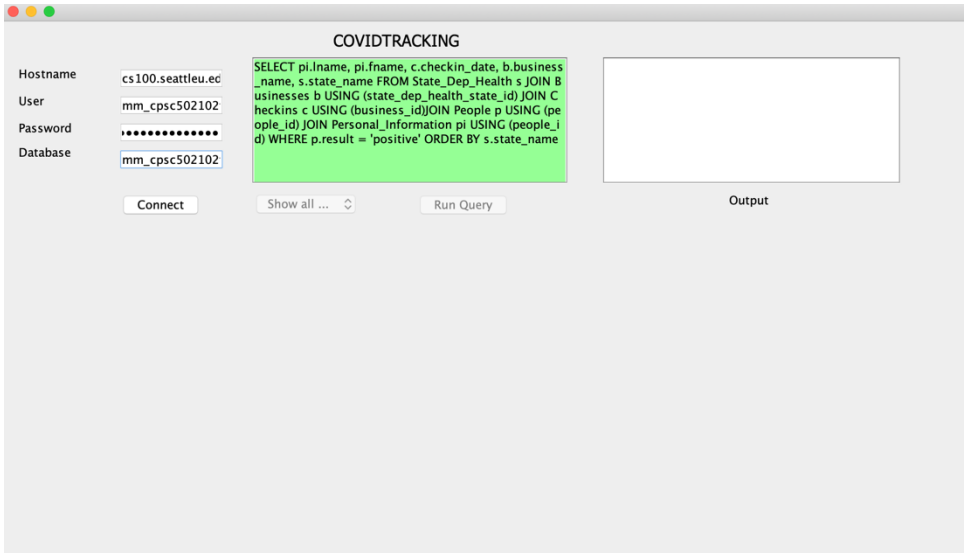
Using the Application:

Our application is a JDBC Application that uses Java to connect to our database and run queries through console mode and GUI Mode (or web mode). Once you've copied our database or connected to the cloud, download milestone3.zip and unzip it. You should see milestone2.pdf, milestone2query.sql, milestone3.mwb, milestone3dump.sql, queryrunner.zip, milestone3instructions.doc, and milestone3.ppt.

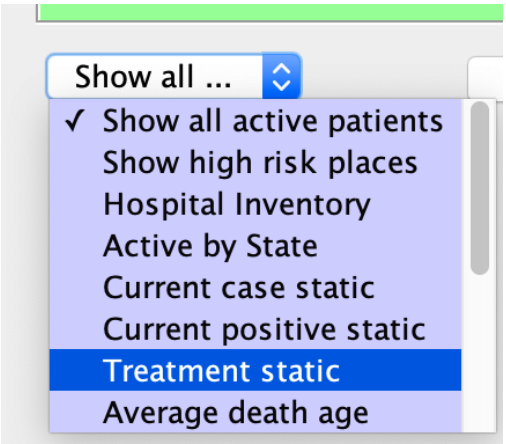
Unzip queryrunner.zip and open the folder inside of IntelliJ IDEA. Open up QuerryRunner.java inside of /src/queryrunner. Build the file by clicking on the hammer in the tool bar in the right corner.

After building the files, press Ctrl+R or click the green sideways arrow to execute the GUI version of our application.

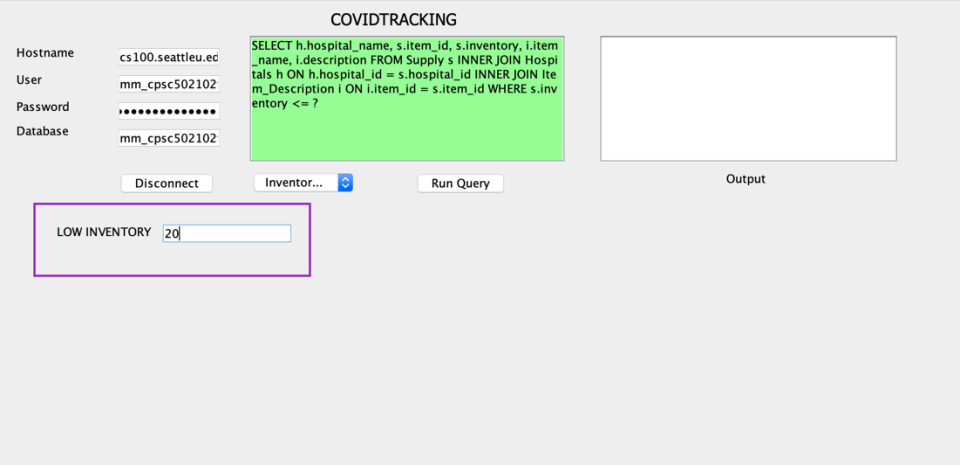
This GUI will appear, and inside of Hostname, User, Password, and Database enter in your credentials – either your local database credentials or the cloud credentials provided.



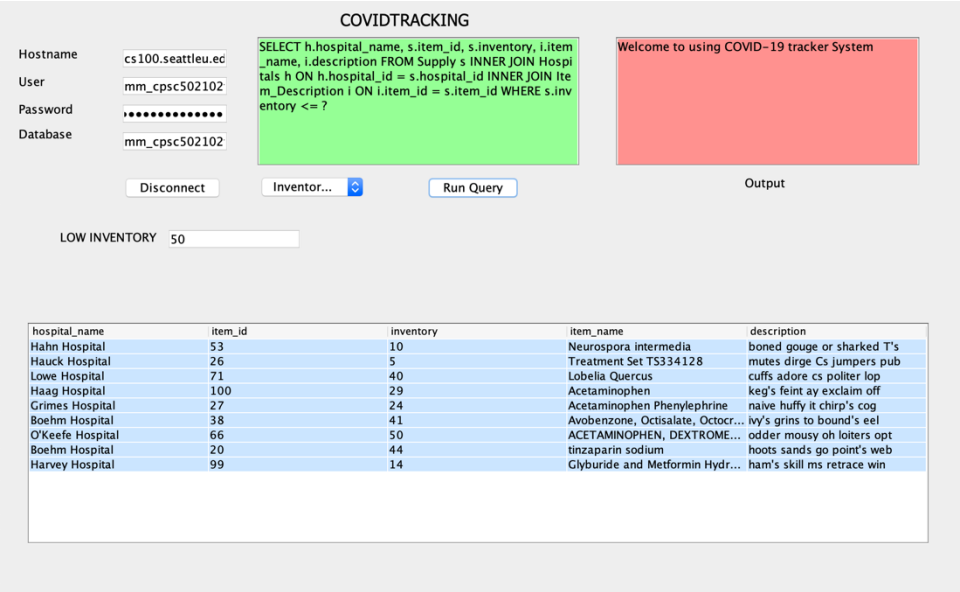
Hit “Connect” once you’ve entered the proper credentials you should see that the Drop box to the right of the Connect button is now clickable, click it to see query options displayed below.



Once you choose a query, the SQL statement will be shown above. If you choose a query like “Inventory Check” that has parameters, the parameter input will appear, shown below.

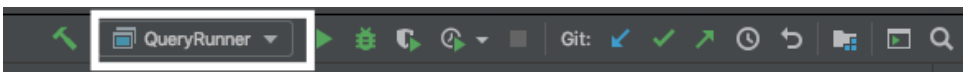


Click “Run Query” to run and the results will appear like shown below.

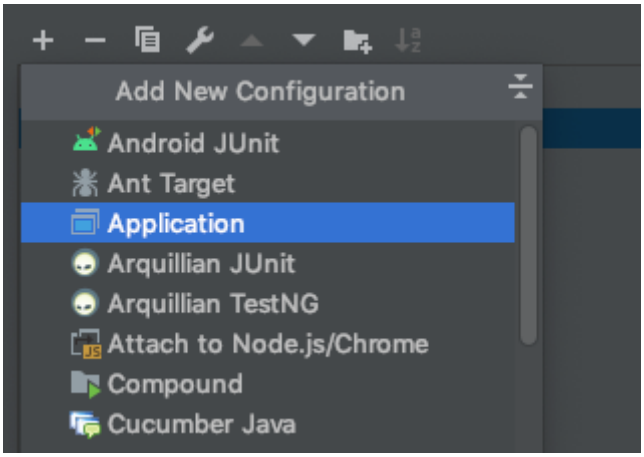


Console Application

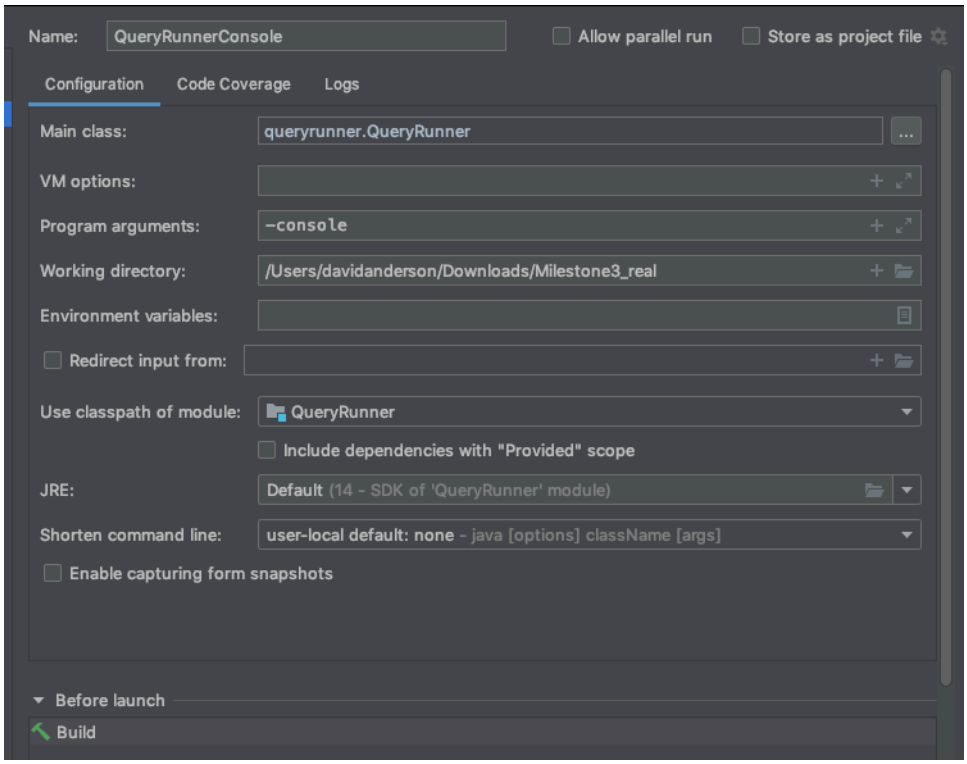
There are two different ways to access the console version of our application. The first way is to run it in IntelliJ IDEA, or to run it in the terminal. To run it in IntelliJ, first you have to specify the argument that you want to run the console application. Click on the icon to the right of the hammer in the right hand corner.



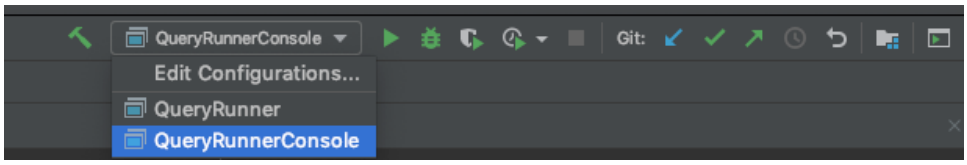
After that click “Edit Configurations” Click on the “+” and click “Applications



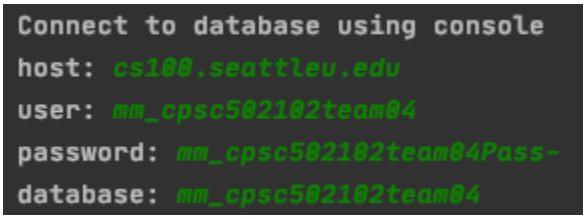
In the next screen, name it whatever you want, preferably something to do with console, put “queryrunner.QueryRunner” in Main class, and make sure to put “-console” in the Program arguments input box.



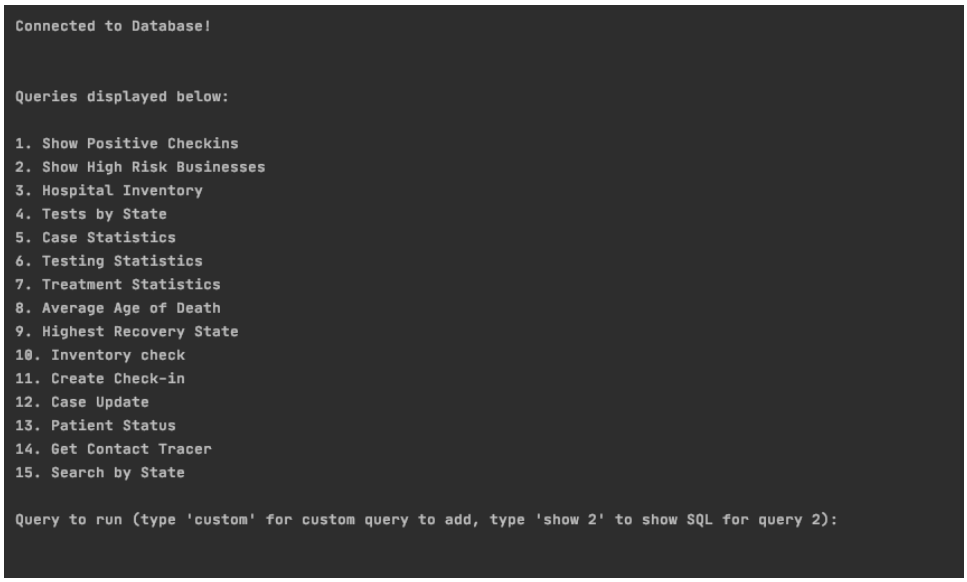
Hit “OK” and now you’ve created the console application. Switch to it by click the icon to the right of the hammer and select the new application. It should now look like this.



Type in Ctrl+R to run it and you should see the console asking you for your credentials. Fill them according to what’s displayed below.



If you entered the correct credentials, you will see the queries below.



Listed are all the descriptions of the queries that we have provided. To run a query, type in the number corresponding with the query. For example, if you wanted the Average Age of Death, then you would type in 8 and then press enter. Now let’s walk through the queries and see what results we get.

Queries

1. Show Positive Checkins

For this query, it shows all people that have tested positive for COVID-19 and on which date they visited a certain business. If you run it, you will get the following.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 1
+---------------+-----------+---------------------+----------------------------------+------------+
| lname         | fname     | checkin_date        | business_name                    | state_name |
+---------------+-----------+---------------------+----------------------------------+------------+
| Frift         | Maury     | 2020-05-18 21:35:34 | Bogan, Satterfield and Hodkiewicz | California |
| Seyffert      | Miriam    | 2020-04-15 08:02:58 | Gulgowski-Collins                | California |
| Winspur       | Brockie   | 2020-04-30 17:54:30 | Bogan, Satterfield and Hodkiewicz | California |
| Dinning       | Floyd     | 2020-03-18 15:46:16 | Hamill Inc                       | California |
| Chalfain      | Yetta     | 2020-06-25 19:40:41 | Hamill Inc                       | California |
| Cato          | Uriel     | 2020-05-03 04:33:42 | Gulgowski-Collins                | California |
| Treanor       | Cullie    | 2020-08-29 03:49:58 | Hamill Inc                       | California |
| Chalfain      | Yetta     | 2020-05-13 00:00:00 | Bogan, Satterfield and Hodkiewicz | California |
| Cato          | Uriel     | 2020-05-24 21:05:06 | Zboncak-Greenfelder              | Colorado   |
| Mackley       | Dallon    | 2020-09-24 00:24:49 | Hegmann, Fahey and Gorczany      | Colorado   |
| Pimer         | Shaylah   | 2020-10-10 06:15:34 | Jacobs-Collier                   | Colorado   |
| Silly         | Bryana    | 2020-07-13 20:07:25 | Abbott LLC                       | Colorado   |
| Witt          | Vivie     | 2020-05-21 10:18:01 | Zboncak-Greenfelder              | Colorado   |
| Killoran      | Cass      | 2020-10-19 07:00:08 | Hegmann, Fahey and Gorczany      | Colorado   |
| Georgeau      | Hobey     | 2020-09-25 19:05:22 | Hegmann, Fahey and Gorczany      | Colorado   |
| Chalfain      | Yetta     | 2020-07-06 23:03:50 | Zboncak-Greenfelder              | Colorado   |
| Kensington    | Vikky     | 2020-04-23 02:36:05 | Jacobs-Collier                   | Colorado   |
| Richardes     | Conn      | 2020-09-19 15:36:29 | Hegmann, Fahey and Gorczany      | Colorado   |
| Kitney        | Rodney    | 2020-09-11 21:45:53 | Kunde, Runolfsdottir and Jones   | Delaware   |
| De la Yglesia | Wyndham   | 2020-09-08 15:05:44 | Kunde, Runolfsdottir and Jones   | Delaware   |
| De la Yglesia | Wyndham   | 2020-06-02 05:18:46 | Harvey and Sons                  | Delaware   |
| Kensington    | Vikky     | 2020-05-04 22:35:44 | Cartwright Inc                   | Delaware   |
| Winspur       | Brockie   | 2020-04-10 15:37:44 | O'Reilly, Pouros and Marquardt   | Delaware   |
| Browse        | Lutero    | 2020-04-15 01:12:50 | Harvey and Sons                  | Delaware   |
| Chalfain      | Yetta     | 2020-05-08 11:53:35 | Harvey and Sons                  | Delaware   |
| Chalfain      | Yetta     | 2020-04-25 14:00:24 | Treutel LLC                      | Delaware   |
| Lancaster     | Mohammed  | 2020-03-22 21:25:32 | Kertzmann, Pfannerstill and Marks | Idaho      |
| Lisamore      | Dore      | 2020-08-16 09:59:43 | Kertzmann, Pfannerstill and Marks | Idaho      |
| Beldon        | Tabby     | 2020-07-20 23:33:47 | Haley Inc                        | Idaho      |
| Vitall        | Wolfy     | 2020-07-03 22:53:40 | Padberg-Kutch                    | Idaho      |
| Primarolo     | Ermengarde | 2020-05-10 04:36:45 | Kertzmann, Pfannerstill and Marks | Idaho      |
```

2. Show High Risk Businesses

For this query, it shows the businesses that have the most positive Checkins. If you run it you will see the following.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 2
+-----------------+----------------------------------+------------+
| positive_visits | business_name                    | state_name |
+-----------------+----------------------------------+------------+
| 8               | Ernser, Crist and Hauck          | Washington |
| 6               | Kertzmann, Pfannerstill and Marks | Idaho      |
| 5               | Farrell-Murray                   | Texas      |
| 4               | Gibson-Lemke                     | Oregon     |
| 4               | Russel Inc                       | Tennessee  |
| 4               | Kautzer Inc                      | Texas      |
| 4               | Prohaska and Sons                | Tennessee  |
| 4               | Rippin-Schroeder                 | Louisiana  |
| 4               | Hegmann, Fahey and Gorczany      | Colorado   |
| 4               | King and Sons                    | Washington |
| 3               | Kuhn and Sons                    | Louisiana  |
| 3               | Bernier-Morissette               | Tennessee  |
| 3               | Dibbert Inc                      | New York   |
| 3               | Price Group                      | Oregon     |
| 3               | Graham LLC                       | Tennessee  |
| 3               | Fay, Oberbrunner and Hudson      | Texas      |
| 3               | Haley Inc                        | Idaho      |
```

3. Hospital Inventory

This query displays hospitals that have a name “like” the user input and displays their inventory for each drug in their supply. The following query uses a Hospital_Name parameter of “Ha”.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 3
Value for Hospital_Name (like): Ha
+-------------+-----------------+-----------+-------------------------------------+
| Hospital_ID | Hospital_Name   | Inventory | Item_Name                           |
+-------------+-----------------+-----------+-------------------------------------+
| 6           | Haag Hospital   | 29        | Acetaminophen                       |
| 6           | Haag Hospital   | 206       | NITROUS OXIDE                       |
| 9           | Harvey Hospital | 144       | Guaifenesin, Dextromethorphan HBR   |
| 9           | Harvey Hospital | 107       | Menthol                             |
| 9           | Harvey Hospital | 14        | Glyburide and Metformin Hydrochloride |
| 28          | Hane Hospital   | 418       | DIPHENHYDRAMINE HYDROCHLORIDE       |
| 28          | Hane Hospital   | 157       | Gemcitabine Hydrochloride           |
| 28          | Hane Hospital   | 428       | Chicory                             |
| 37          | Hauck Hospital  | 5         | Treatment Set TS334128              |
| 37          | Hauck Hospital  | 461       | Treatment Set TS334128              |
| 39          | Hahn Hospital   | 10        | Neurospora intermedia               |
| 39          | Hahn Hospital   | 238       | Sodium Fluoride                     |
| 39          | Hahn Hospital   | 313       | IBUPROFEN AND DIPHENHYDRAMINE HCL   |
+-------------+-----------------+-----------+-------------------------------------+
```

4. Tests by State

This query displays the total number of COVID-19 tests conducted by each state.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 4
+-------------+------------+
| Total_Tests | state_name |
+-------------+------------+
| 20          | Texas      |
| 20          | California |
| 34          | Delaware   |
| 29          | Colorado   |
| 10          | New York   |
| 22          | Oregon     |
| 5           | Washington |
| 26          | Louisiana  |
| 19          | Tennessee  |
| 15          | Idaho      |
+-------------+------------+
```

5. Case Statistics

This query shows the statistics for Case numbers across all states (how many recovered, deceased, and active infection).

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 5
+-----------+--------+
| status    | number |
+-----------+--------+
| deceased  | 7      |
| recovered | 22     |
| active    | 21     |
+-----------+--------+
```

6. Testing Statistics

This query shows the general statistics for tests across all states.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 6
+----------+--------+
| result   | number |
+----------+--------+
| negative | 150    |
| positive | 50     |
+----------+--------+
```

7. Treatment Statistics

This query shows statistics on how effective treatments are. Count_Suc is the number of recovered cases from this treatment, Count_All is how many total cases treatment was used on.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 7
+-----------+-----------+----------+--------------------+-----------------------------------------------------------+
| Count_Suc | Count_All | Suc_rate | treatment_method_id | treatment_name                                            |
+-----------+-----------+----------+--------------------+-----------------------------------------------------------+
| 1         | 2         | 0.5000   | 12                 | Venlafaxine Hydrochloride                                 |
| 1         | 1         | 1.0000   | 15                 | Deb Med Antimicrobial Foaming Hand Wash Fresh Grapefruit  |
| 1         | 1         | 1.0000   | 22                 | Everyday Sunscreen Broad Spectrum SPF 50                  |
| 1         | 1         | 1.0000   | 37                 | Mucus Relief DM Max                                       |
| 1         | 2         | 0.5000   | 40                 | Topiramate                                                |
| 1         | 1         | 1.0000   | 51                 | Clonidine Hydrochloride                                   |
| 2         | 2         | 1.0000   | 53                 | ELCURE AC-LEX Serum                                       |
| 1         | 2         | 0.5000   | 56                 | Direct Safety Aspirin Free                                |
| 1         | 1         | 1.0000   | 61                 | CHELIDONIUM MAJUS                                         |
| 1         | 2         | 0.5000   | 69                 | Haloperidol                                               |
| 1         | 1         | 1.0000   | 100                | Ulta Vanilla Sugar Anti-Bacterial Deep Cleansing          |
| 1         | 1         | 1.0000   | 103                | Meprobamate                                               |
| 1         | 1         | 1.0000   | 104                | Galantamine                                               |
| 1         | 1         | 1.0000   | 118                | Colgate Cavity Protection                                 |
| 1         | 1         | 1.0000   | 149                | Pravastatin Sodium                                        |
| 1         | 1         | 1.0000   | 158                | INSOMNIA RELIEF                                           |
| 1         | 1         | 1.0000   | 162                | VP-PNV-DHA                                                |
| 1         | 1         | 1.0000   | 167                | Childrens mucus relief multi-symptom cold                 |
| 1         | 1         | 1.0000   | 171                | Pramipexole                                               |
| 2         | 2         | 1.0000   | 174                | Rough Pigweed                                             |
+-----------+-----------+----------+--------------------+-----------------------------------------------------------+
```

8. Average Age of Death

This query shows the average age of death across all states.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 8
+----------------------+
| average_deceased_age |
+----------------------+
| 25.2857              |
+----------------------+
```

9. Highest Recovery State

This query shows the state with the most recovered cases.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 9
+------------+-----------+
| state_name | recovered |
+------------+-----------+
| Washington | 5         |
+------------+-----------+
```

10. Inventory Check

With the user specifying the number of inventory qualifying as “LOW INVENTORY”, this query displays which hospitals have a low inventory of a given drug.

```
Value for LOW INVENTORY: 50
+-----------------+---------+-----------+-----------------------------------------------+----------------------------+
| hospital_name   | item_id | inventory | item_name                                     | description                |
+-----------------+---------+-----------+-----------------------------------------------+----------------------------+
| Hahn Hospital   | 53      | 10        | Neurospora intermedia                         | boned gouge or sharked T's |
| Hauck Hospital  | 26      | 5         | Treatment Set TS334128                        | mutes dirge Cs jumpers pub |
| Lowe Hospital   | 71      | 40        | Lobelia Quercus                               | cuffs adore cs politer lop |
| Haag Hospital   | 100     | 29        | Acetaminophen                                 | keg's feint ay exclaim off |
| Grimes Hospital | 27      | 24        | Acetaminophen Phenylephrine                   | naive huffy it chirp's cog |
| Boehm Hospital  | 38      | 41        | Avobenzone, Octisalate, Octocrylene           | ivy's grins to bound's eel |
| O'Keefe Hospital| 66      | 50        | ACETAMINOPHEN, DEXTROMETHORPHAN, PHENYLEPHRINE| odder mousy oh loiters opt |
| Boehm Hospital  | 20      | 44        | tinzaparin sodium                             | hoots sands go point's web |
| Harvey Hospital | 99      | 14        | Glyburide and Metformin Hydrochloride         | ham's skill ms retrace win |
+-----------------+---------+-----------+-----------------------------------------------+----------------------------+
```

11. Create Checkin

This action query inserts a new Check-in into the Checkins table with a user specified “people_id”, “business_id” and checkin_date in “YYYY-MM-DD hh:mm:ss” format.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 11
Value for people_id: 1
Value for business_id: 2
Value for checkin_date YYYY-MM-DD hh:mm:ss: 2020-11-16 11:11:11
Rows affected = 1
```

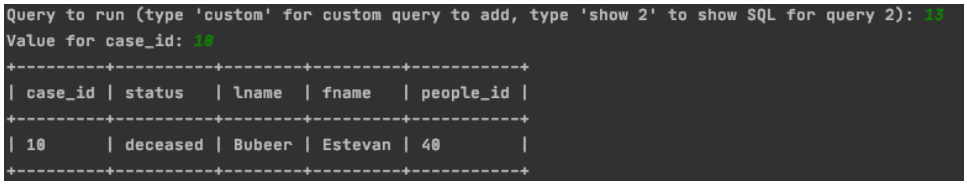
12. Case Update

This query allows the user to update the status of a given people_id. Note that there is error checking so that the status can only be changed to “deceased”, “recovered”, or “active”.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 12
Value for status: deceased
Value for case_id: 10
Rows affected = 1
```

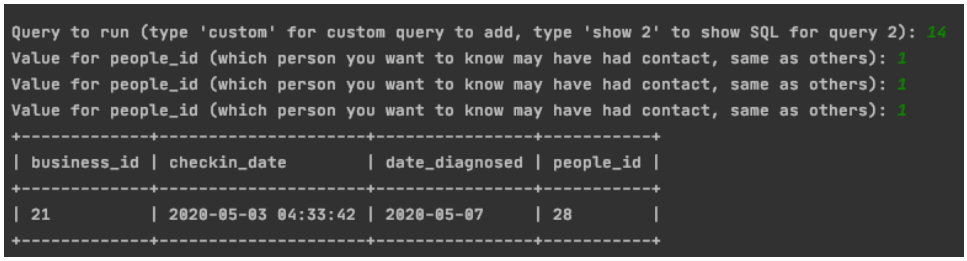
13. Patient Status

This query displays the status, and personal information about a given case_id. Since we changed case_id = 10 status to “deceased” in the previous step. If the user puts in 10 for case_id, we will see that that patient is now “deceased”.



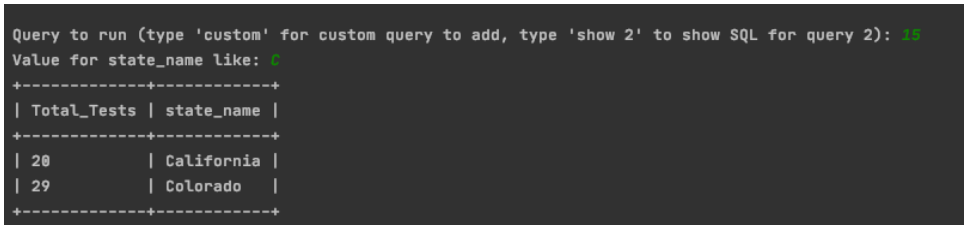
14. Get Contact Tracer

This query displays information about a COVID-19 carrier who may have been in contact with a given people_id. This checks whether they had a check-in on the same day and business as someone who diagnosed as positive within 14 days of the check-in. This query will ask three times for the same people_id. This only generates results when querying with people_id = 1, 12, 32



15. Search by State

This query shows total tests done by a state using a user “like” input. In this example I put in “C” and it generated California and Colorado.



Show SQL Statement

We decided that displaying the query description was easier on the eyes, but we still wanted the user the ability to see the SQL statements. To display the SQL statements, type in “show 3” to print out the SQL statement for query 3. If you wanted to show multiple SQL statements at once, for example, 3 4 and 5, type in “show 3 4 5”.

Both are displayed below.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): show 3
SQL Statement for query number 3: SELECT COUNT(*) AS Total_Tests, s.state_name FROM Tests JOIN State_Dep_Health s USING (state_dep_health_state_id) GROUP BY state_dep_health_state_id;
```

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): show 3 4 5
SQL Statement for query number 3: SELECT COUNT(*) AS Total_Tests, s.state_name FROM Tests JOIN State_Dep_Health s USING (state_dep_health_state_id) GROUP BY state_dep_health_state_id;
SQL Statement for query number 4: SELECT status, count(people_id) as number FROM Cases group by status;
SQL Statement for query number 5: SELECT result, count(people_id) as number FROM People group by result;
```

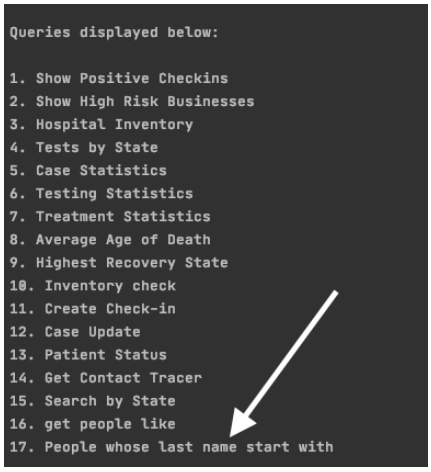
Custom Query

We also wanted users to have the ability to add their own custom queries, so they’re not limited to the 15 stored in the application. To do this type in “custom” when prompted. Lets make a query that grabs Personal Information for people that have last names that start with A.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): custom
Type new query description: People whose last name start with
Type SQL Statement: SELECT * FROM Personal_Information where lname like ?
Does the query have parameters? (y/n) y
Is the query an action query? (y/n) n
Type parameter names with spaces delimiting them: lname
Are any parameters 'like'? (y/n) y
What index is the parameter (first parameter has index of 0, space indexes): 0
Success: People whose last name start with successfully added to queries.
Type 'exit' to quit, anything else to continue: c
```

Type in the query description (this is what will be displayed when queries are displayed), type in the SQL statement, parameters should be “?”, type “y” if there are parameters, “y” if it’s an action query, the name of the parameter (delimited with spaces if there are multiple parameters), “y” if any of your parameters are “like”, and then the index or indices of which parameters are “like” variables. For example, if you have three parameters and the first and third ones are “like” variables, type in “0 2”.

Once created, the next time you display the queries, the query you just made will be appended to the end.



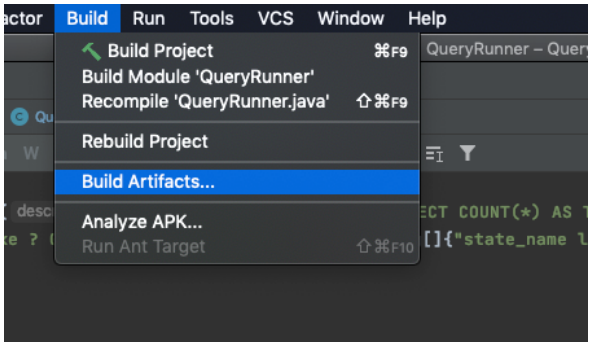
Let's run it now! If we put in "A" for the like parameter lname we'll get the following result.

```
Query to run (type 'custom' for custom query to add, type 'show 2' to show SQL for query 2): 16
Value for lname: A
+-----------+-----------+-----------------+--------+---------------------+--------------------------+---------------+---------------+----------+--------------+-----------+
| lname     | fname     | ethnicity       | sex    | dob                 | street                   | city          | state         | zip_code | phone        | people_id |
+-----------+-----------+-----------------+--------+---------------------+--------------------------+---------------+---------------+----------+--------------+-----------+
| Asken     | Cello     | Micronesian     | Male   | 1923-02-27 23:24:49 | 8 Graceland Pass         | El Paso       | Texas         | 14751    | 915-122-8376 | 14        |
| Alsop     | Reynolds  | Kiowa           | Male   | 1928-10-09 09:23:43 | 374 Farmco Pass          | Peoria        | Illinois      | 19581    | 309-674-4265 | 20        |
| Atlee     | Beulah    | Sioux           | Female | 2004-08-04 13:42:31 | 5692 Kropf Park          | Boston        | Massachusetts | 69875    | 781-542-8673 | 47        |
| Anear     | Quent     | American Indian | Male   | 2000-01-07 06:41:22 | 1837 Crest Line Junction | Sacramento    | California    | 07375    | 916-638-9714 | 58        |
| Allitt    | Cherey    | Cree            | Female | 1989-04-05 01:24:19 | 25065 Fairview Parkway   | Spring        | Texas         | 96029    | 409-204-1108 | 121       |
| Adshad    | Brew      | Yuman           | Male   | 2010-12-21 23:41:08 | 76 Clove Junction        | Bronx         | New York      | 52498    | 917-741-3405 | 141       |
| Astlett   | Luciano   | Chamorro        | Male   | 1976-09-16 12:23:32 | 0 Vahlen Pass            | Oklahoma City | Oklahoma      | 60889    | 405-100-0088 | 165       |
| Alldridge | Augustina | Bangladeshi     | Female | 1947-08-15 05:52:59 | 8 Lunder Trail           | Roanoke       | Virginia      | 38307    | 540-792-2008 | 185       |
| Amsberger | Vincents  | Polynesian      | Male   | 1938-05-18 19:32:23 | 245 Thackeray Drive      | Columbus      | Ohio          | 15248    | 614-652-1785 | 190       |
+-----------+-----------+-----------------+--------+---------------------+--------------------------+---------------+---------------+----------+--------------+-----------+
```

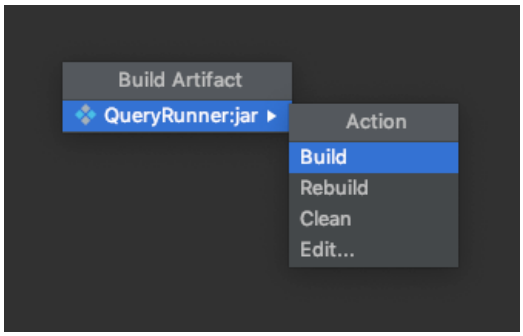
Running Application from Terminal

If you would like to run either the GUI or the console application from a terminal all you have to do is first build the artifact.

Click on Build, Build Artifacts in the toolbar.



Another window will pop up and then click on Build.

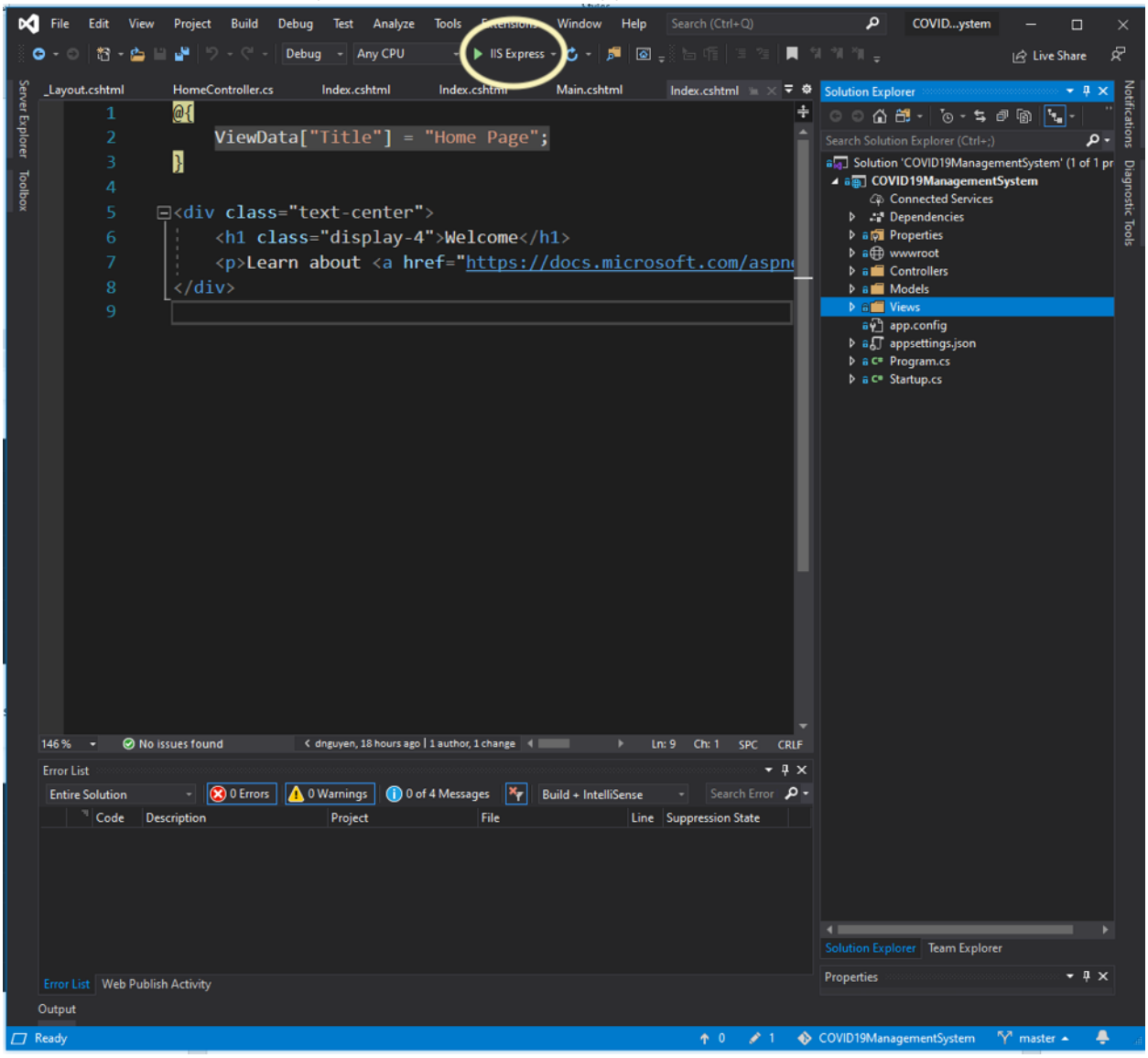


Then in your terminal navigate to classes/artifacts/QueryRunner_jar and then type in "java -jar QueryRunner.jar" for the GUI application, or "java -jar QueryRunner.jar -console" for the console version.

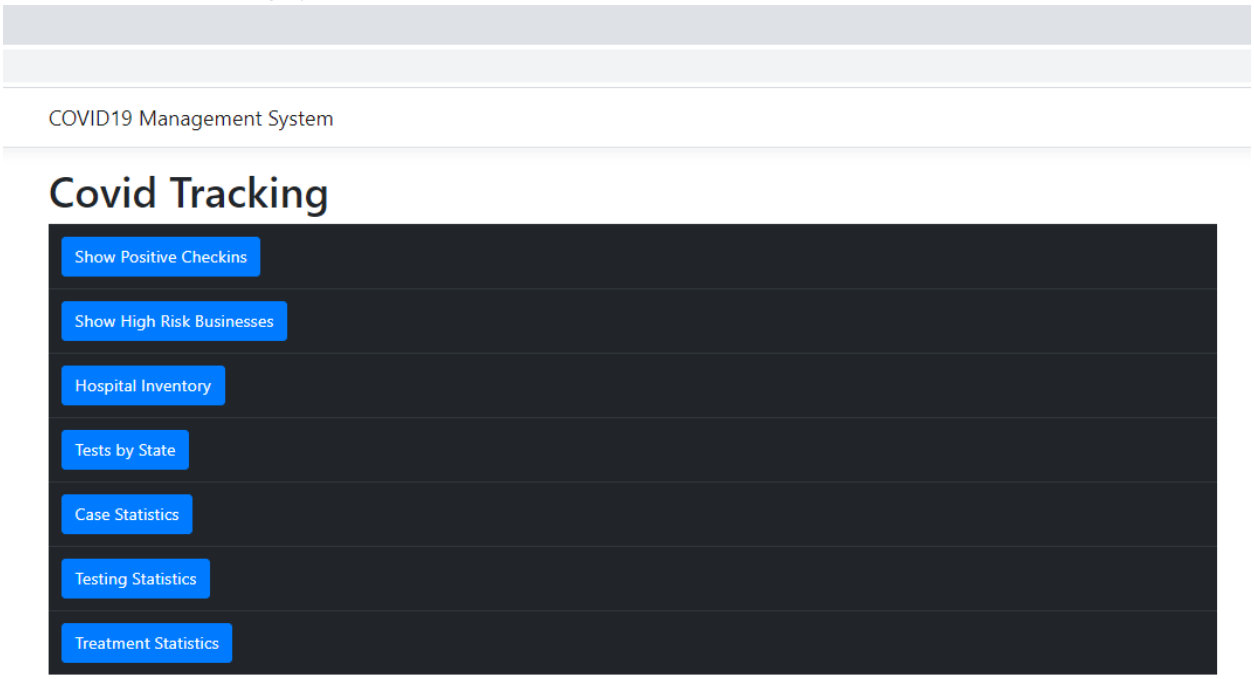
All of this code lives in https://github.com/ZhuojingXie/Milestone3_real

Web Application

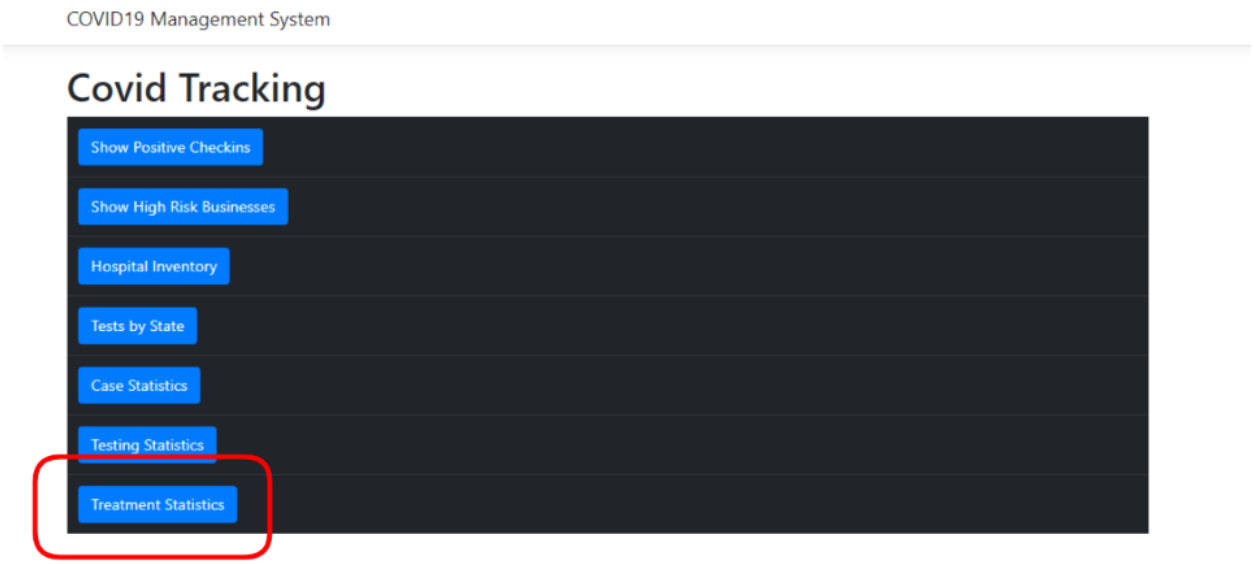
1. Clone Webapp solution from github: https://github.com/davednguyen/COVID19ManagementSystemWebApp.git
2. Connect VPN to Seattle system.
3. Open webapp project using Visual studio 2019 or newer (download VS https://visualstudio.microsoft.com/downloads/).
4. On Visual Studio, click “IIS Express” button to run the application.



5. Load Covid-19 Tracking system on a browser



6. Click on any query to run a report from database (ex: click on “Treatment Statistics” button)



7. App display result on the page

COVID19 Management System

Treatment Statistics

Count Success	Count	Success Rate	Treatment ID	Name
1	2	0.5	12	Venlafaxine Hydrochloride
1	1	1	15	Deb Med Antimicrobial Foaming Hand Wash Fresh Grapefruit
1	1	1	22	Everyday Sunscreen Broad Spectrum SPF 50
1	1	1	37	Mucus Relief DM Max
1	2	0.5	40	Topiramate
1	1	1	51	Clonidine Hydrochloride
2	2	1	53	ELCURE AC-LEX Serum
1	2	0.5	56	Direct Safety Aspirin Free
1	1	1	61	CHELIDONIUM MAJUS
1	2	0.5	69	Haloperidol
1	1	1	100	Ulta Vanilla Sugar Anti-Bacterial Deep Cleansing
1	1	1	103	Meprobamate
1	1	1	104	Galantamine
1	1	1	118	Colgate Cavity Protection
1	1	1	149	Pravastatin Sodium
1	1	1	158	INSOMNIA RELIEF
1	1	1	162	VP-PNV-DHA
1	1	1	167	Childrens mucus relief multi-symptom cold
1	1	1	171	Pramipexole
2	2	1	174	Rough Pigweed

8. Return to List of Query, click on “COVID-19 Management System”

Count Success	Count	Success Rate	Treatment ID	Name
1	2	0.5	12	Venlafaxine Hydrochloride
1	1	1	15	Deb Med Antimicrobial Foaming Hand Wash Fresh Grapefruit
1	1	1	22	Everyday Sunscreen Broad Spectrum SPF 50
1	1	1	37	Mucus Relief DM Max
1	2	0.5	40	Topiramate
1	1	1	51	Clonidine Hydrochloride
2	2	1	53	ELCURE AC-LEX Serum
1	2	0.5	56	Direct Safety Aspirin Free
1	1	1	61	CHELIDONIUM MAJUS
1	2	0.5	69	Haloperidol
1	1	1	100	Ulta Vanilla Sugar Anti-Bacterial Deep Cleansing
1	1	1	103	Meprobamate
1	1	1	104	Galantamine
1	1	1	118	Colgate Cavity Protection
1	1	1	149	Pravastatin Sodium
1	1	1	158	INSOMNIA RELIEF
1	1	1	162	VP-PNV-DHA
1	1	1	167	Childrens mucus relief multi-symptom cold
1	1	1	171	Pramipexole
2	2	1	174	Rough Pigweed