Ori Chanael

Donner Hanson

Rene German

CPSC 408

5/22/2020

Final Project Report

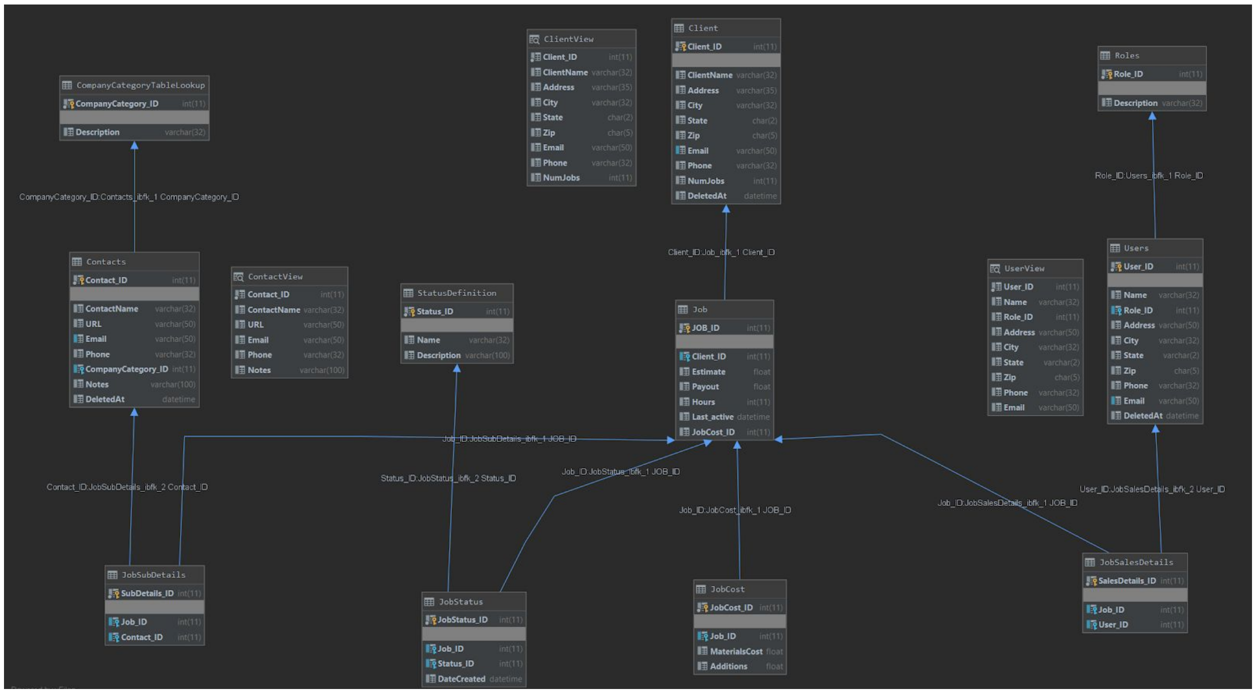
Introduction

What became our final project was not our first idea. Originally, we wanted to create a database system that would interact with one of our favorite games, Destiny 2 by BUNGIE. Destiny allows players to organize themselves in ‘clans,’ player created groups that offer certain rewards, but is mostly for the ease of communication for activities that were not ‘matchmade’ (entering a queue with random players). The idea was to use the API provided by BUNGIE and scrape the data from it to compile our database. This application would allow for clan moderators to add, remove, and organize other members. More normal members could see their statistics from playtime.

It quickly became apparent however, that we did not have the pre-requisite experience in networking to successfully integrate APIs to our project during the time allotted for the assignment. We preferred to bring forth a fully functional database that could perform all the tasks specified by the project guidelines, be secure to user input, and be easily integrated with a UI in the future and be useful in a real-world industry. We settled on a back-up plan: a contractor management application.

Ori’s father works as a contractor in the construction industry, which is where the inspiration arose. Currently, he uses DialersPro, an integrated telemarketer and database management system to organize jobs, clients, and salespeople, while all financial concerns are handled via Quickbooks. Ori conducted research by having face to face communication with his father and various admins about the current systems they use and some of the pitfalls of those applications. We both then decided the best way to organize the necessary data and tables by drawing out diagrams and conversing about project scope and deadlines to have features created. The current state of the application is designed to easily manage various jobs, employees, clients, and subcontractor contacts and we are planning on expanding the abilities and UI of the application over the next few months. Currently, the application allows a user to create, update, and delete various records needed to run such a business. The database itself is created in a way so that it is easy to refer to and gather information across multiple tables by combining functional programming in Python with MySQL using Google Cloud Platform.

Overview



Presented above is the schema diagram for the database. The root table is the Job table. From the Job table, any other table can be accessed through a series of foreign keys that correspond to the primary key of a connecting table. For example, the Job and Contacts tables can be easily joined via the JobSubDetails table, which contains foreign keys to both tables. This setup makes table joins very smooth and easier to both conceptualize and write. Additionally placed within the diagram are the Views for the Contacts, Users, and Clients tables. These Views allowed us to display these tables without their 'DeletedAt' columns without laboriously typing out each Select statement every time we wished to show these tables.

Each table has an indexed primary key. In addition to the indexed primary key parameter, multiple tables have uniquely indexed email fields. These fields have been indexed because we analyzed a common query pattern of using email, and noted that email addresses MUST be unique to one person. Depending on the compilation and auto-organization of DataGrip's indexing algorithm, indexing can be set up as either a B-Tree or as a hash table. In B-Tree indexing, search, insert, and delete have a Big O run-time of O(log(n)) and have the space of O(n). Hash table indexing has a run-time O(n) in the worst-case scenario for search, insert, delete, and space but in an average case has O(1) for search, insert, and delete and takes the space of O(n).

To increase productivity for often used functionality and queries, and maintain an ACID certified database, we created a large set of procedures to handle inserts and updates into various tables that have built in transaction and rollback features to maintain the integrity of data, in the case that an error occurs during an attempt to commit to the database. Each of these procedures has a structure similar to the following:

```
BEGIN
  DECLARE `_rollback` BOOL DEFAULT 0;
  DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `_rollback` = 1;
  START TRANSACTION;
     INSERT INTO Table(someAttribute)
     VALUES (someInputAttribute);
  IF `_rollback` THEN
     ROLLBACK;
  ELSE
     COMMIT;
  END IF;
END;
```

This format allows us to easily maintain consistency throughout stored procedures, while maintaining the ACID-ity of our data. In a future update to the application we are working to add a log system to the Python application that is linked to our Google Cloud Platform MySQL database to assist in transaction failure monitoring. This will allow us to keep track of the multitude of changes and tests that will undoubtedly emerge in further development of the application.

Features

Currently the ContractorManagementDB application has these easy to use features:

1) Print and display all tables by choice.
2) Useful parameterized search functions to return invaluable data to the user.
3) Record creation amongst relevant tables.
    a) Some tables, like StatusDefintion and Roles, need static records to fulfill their purpose as a way to explain what their IDs mean when attached to their relevant table as a foreign key. Additional roles for

users or statuses for jobs can be added in the future, but for the most part is both not necessary and as needed.

4) Soft deletion of users, contacts, and clients has been implemented by using DateTime objects to remove unwanted records from table views but persist as users may wish to return records back into the database or analyze data of why clients wished to be removed or other invaluable data such as “when users were deleted”.
5) Update Records is available across all pertinent tables, this allows correction of names, emails, costs, and statuses of various jobs and clients.
6) Transactions are implemented within stored procedures for ease of use across all updates and insertions.
7) We currently have an export option which exports data in a structured CSV format for use within other data analysis software like Excel.
    a) We plan to implement parameterized exporting of CSV files such as:
        i) All jobs between certain dates.
        ii) Report of jobs by sales representative.
8) Foreign keys and other constraints to preserve Third Normal Form structuring of the database.
9) String Formatting and parameter parsing in Python to avoid SQL injection and validate input.

Dependencies:

Our application is dependent on the 3rd party libraries Mysql-connector, Pandas, and Faker. Mysql-connector is used to establish a connection with our Google Cloud Platform database. Pandas is used to parse the information retrieved from MySQL into a human-readable, “pretty”, format. Faker is used as a testing utility for generating large amounts of fake data to load into the database.

In a future update the application will be dependent on Kivy, which we have been experimenting with as our UI and learning as we will be implementing that feature over the summer of 2020.

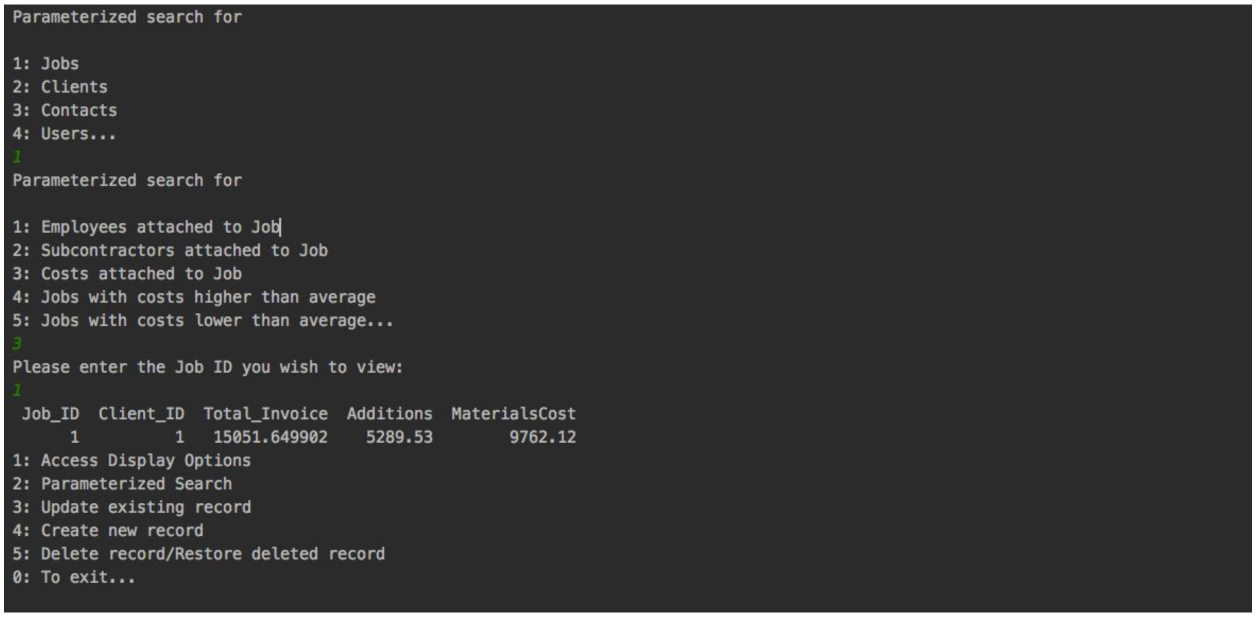
Future Additions and Goals:

We plan on adding various new features into our system over the summer of 2020. We have our eyes set on having a fully functional and clean UI, customizable parameterized search options, logging for database administrators, user verification system, and other changes to make the application as intuitive and easy to use as possible, while not losing any of its flexibility or strength.

We would also like to develop either a companion application, or an added feature to the current app, that tracks financial information regarding the associated database.

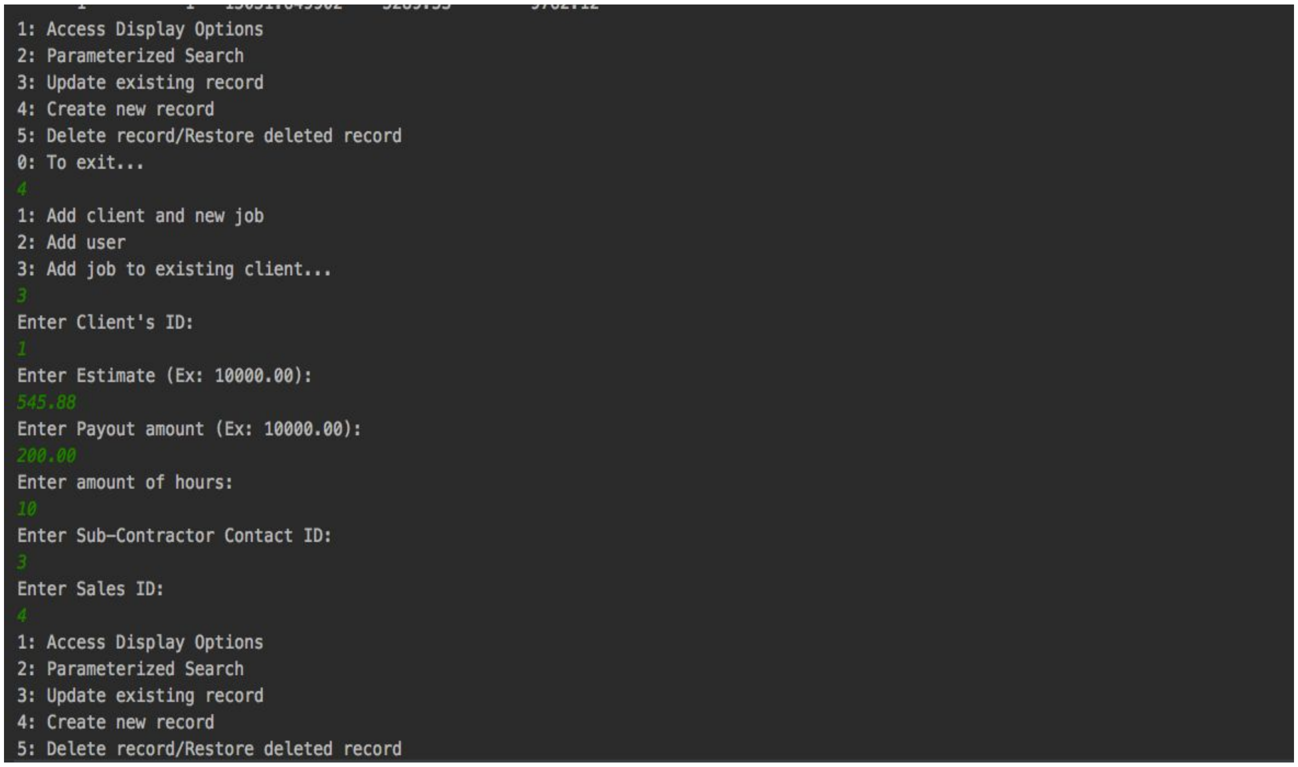
Various Screenshots of Results:

Parameterized Search to display invoices and costs



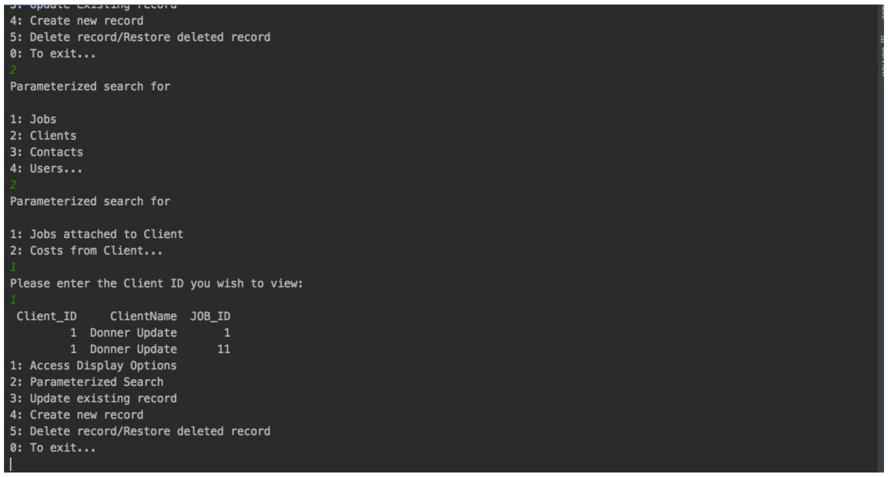
```
Parameterized search for

1: Jobs
2: Clients
3: Contacts
4: Users...
1
Parameterized search for

1: Employees attached to Job
2: Subcontractors attached to Job
3: Costs attached to Job
4: Jobs with costs higher than average
5: Jobs with costs lower than average...
3
Please enter the Job ID you wish to view:
1
 Job_ID  Client_ID  Total_Invoice  Additions  MaterialsCost
      1          1   15051.649902    5289.53        9762.12
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```

Add job to existing client



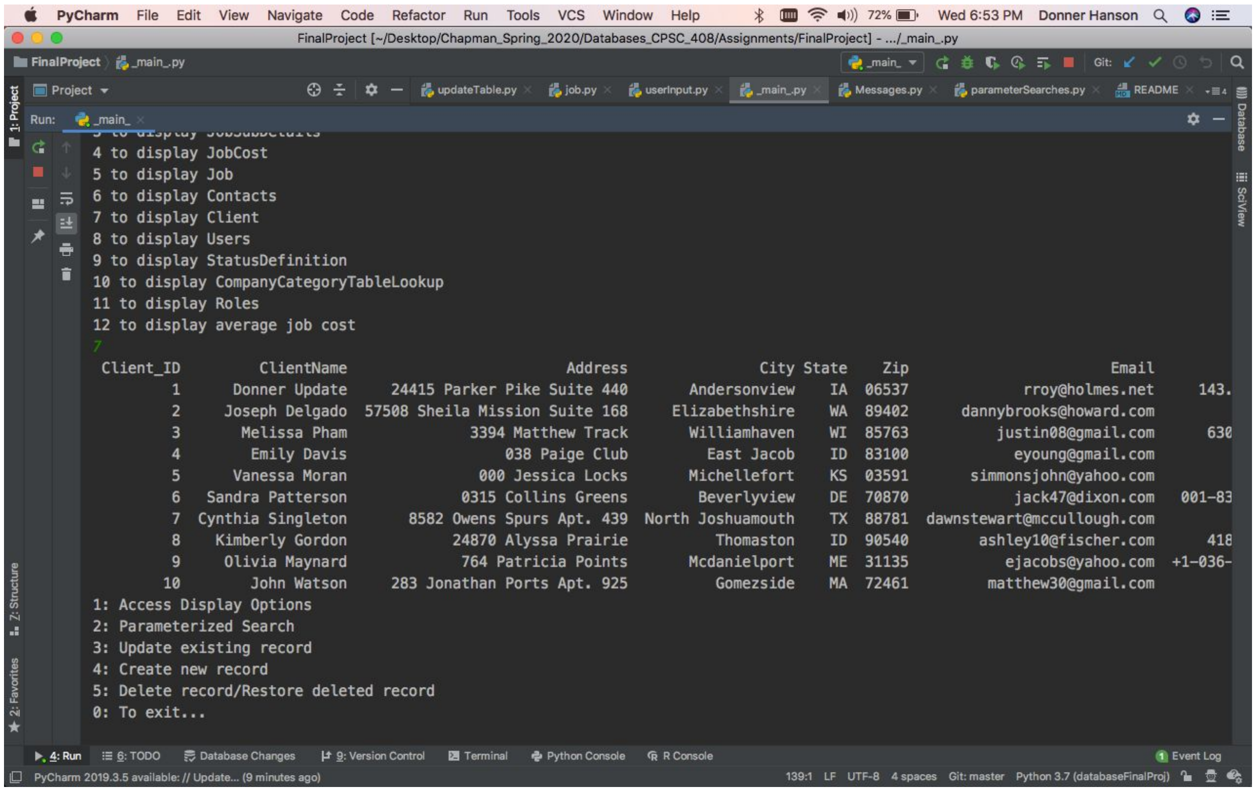
```
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
4
1: Add client and new job
2: Add user
3: Add job to existing client...
3
Enter Client's ID:
1
Enter Estimate (Ex: 10000.00):
545.88
Enter Payout amount (Ex: 10000.00):
200.00
Enter amount of hours:
10
Enter Sub-Contractor Contact ID:
3
Enter Sales ID:
4
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
```

Parameterized search:



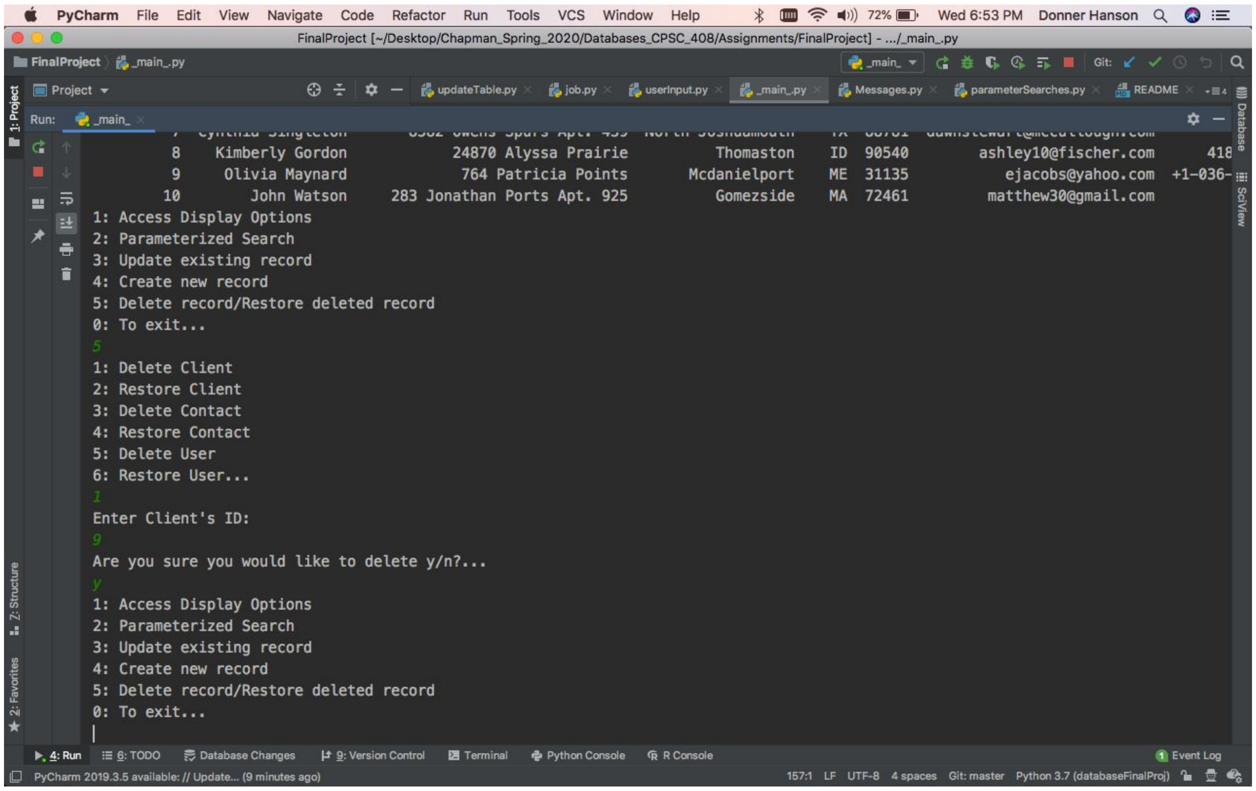
```
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
2
Parameterized search for

1: Jobs
2: Clients
3: Contacts
4: Users...
2
Parameterized search for

1: Jobs attached to Client
2: Costs from Client...
1
Please enter the Client ID you wish to view:
1
 Client_ID     ClientName  JOB_ID
         1  Donner Update       1
         1  Donner Update      11
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```

Full Table Display options:

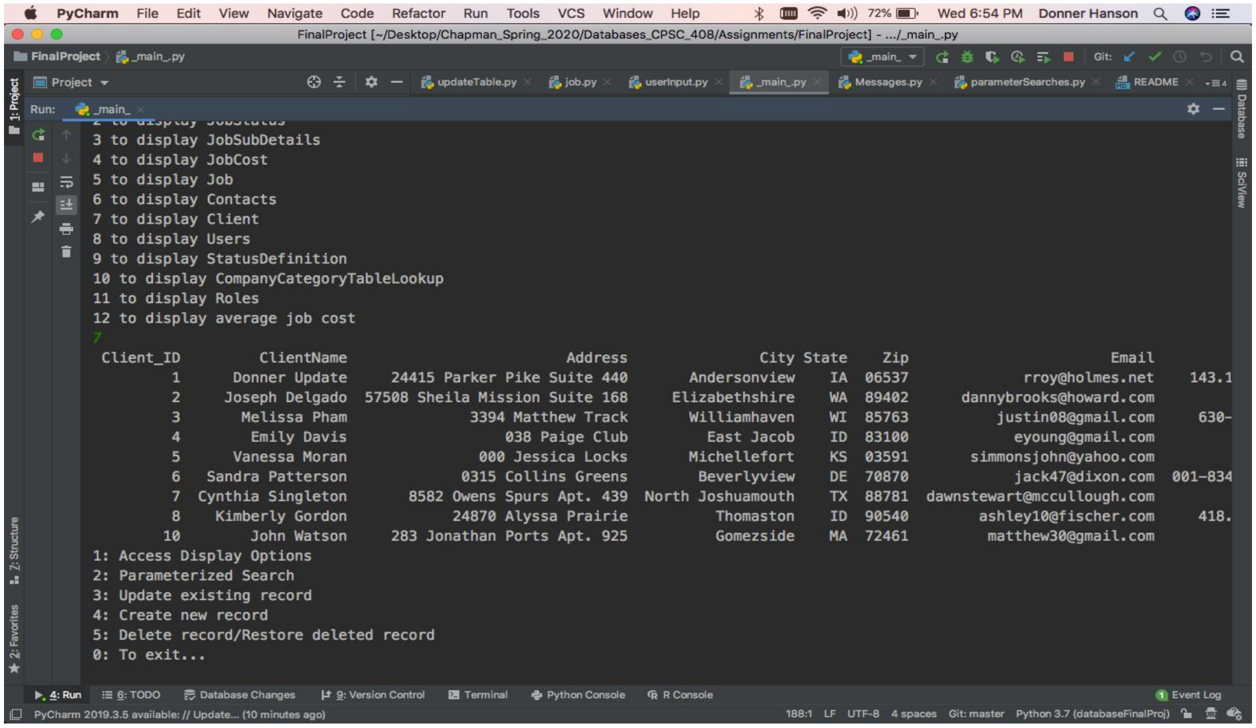


```
4 to display JobCost
5 to display Job
6 to display Contacts
7 to display Client
8 to display Users
9 to display StatusDefinition
10 to display CompanyCategoryTableLookup
11 to display Roles
12 to display average job cost
7
 Client_ID         ClientName                    Address              City State    Zip                      Email
         1      Donner Update    24415 Parker Pike Suite 440     Andersonview    IA  06537             rroy@holmes.net     143.
         2     Joseph Delgado  57508 Sheila Mission Suite 168  Elizabethshire    WA  89402      dannybrooks@howard.com
         3      Melissa Pham             3394 Matthew Track      Williamhaven    WI  85763          justin08@gmail.com      630
         4       Emily Davis                 038 Paige Club        East Jacob    ID  83100           eyoung@gmail.com
         5     Vanessa Moran              000 Jessica Locks      Michellefort    KS  03591     simmonsjohn@yahoo.com
         6   Sandra Patterson             0315 Collins Greens      Beverlyview    DE  70870           jack47@dixon.com   001-83
         7  Cynthia Singleton        8582 Owens Spurs Apt. 439  North Joshuamouth TX  88781  dawnstewart@mccullough.com
         8    Kimberly Gordon            24870 Alyssa Prairie        Thomaston    ID  90540        ashley10@fischer.com      418
         9     Olivia Maynard              764 Patricia Points     Mcdanielport    ME  31135          ejacobs@yahoo.com  +1-036-
        10        John Watson       283 Jonathan Ports Apt. 925       Gomezside    MA  72461         matthew30@gmail.com
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```

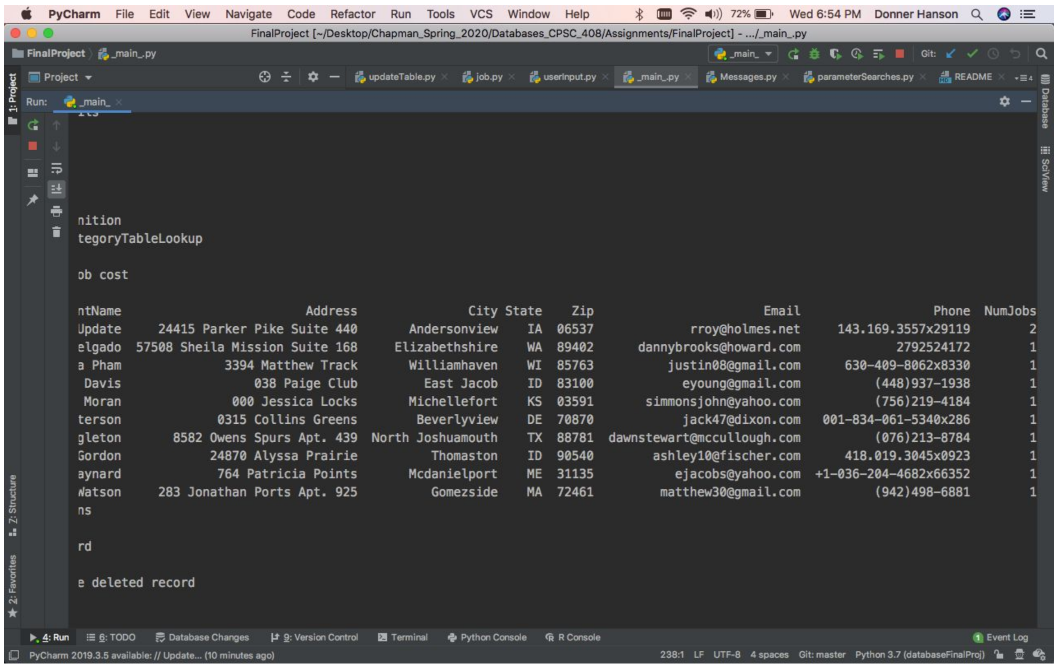
Soft Delete:



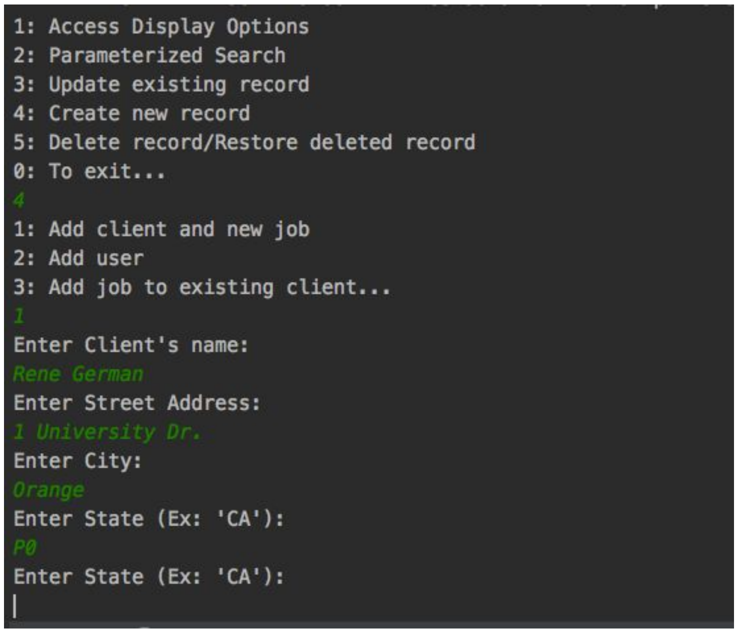
```
         8    Kimberly Gordon          24870 Alyssa Prairie        Thomaston  ID  90540        ashley10@fischer.com      418
         9     Olivia Maynard           764 Patricia Points      Mcdanielport  ME  31135           ejacobs@yahoo.com  +1-036-
        10        John Watson     283 Jonathan Ports Apt. 925      Gomezside  MA  72461         matthew30@gmail.com
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
5
1: Delete Client
2: Restore Client
3: Delete Contact
4: Restore Contact
5: Delete User
6: Restore User...
1
Enter Client's ID:
9
Are you sure you would like to delete y/n?...
y
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```


PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help
FinalProject [~/Desktop/Chapman_Spring_2020/Databases_CPSC_408/Assignments/FinalProject] - .../_main_.py
updateTable.py job.py userInput.py _main_.py Messages.py parameterSearches.py README
Run: _main_
4: Run 6: TODO Database Changes 9: Version Control Terminal Python Console R Console Event Log
PyCharm 2019.3.5 available: // Update... (10 minutes ago)
188:1 LF UTF-8 4 spaces Git: master Python 3.7 (databaseFinalProj)

```
2 to display JobStatus
3 to display JobSubDetails
4 to display JobCost
5 to display Job
6 to display Contacts
7 to display Client
8 to display Users
9 to display StatusDefinition
10 to display CompanyCategoryTableLookup
11 to display Roles
12 to display average job cost
7
 Client_ID        ClientName                     Address             City State   Zip                        Email
         1     Donner Update     24415 Parker Pike Suite 440     Andersonview    IA  06537              rroy@holmes.net    143.1
         2    Joseph Delgado  57508 Sheila Mission Suite 168   Elizabethshire    WA  89402       dannybrooks@howard.com
         3      Melissa Pham            3394 Matthew Track      Williamhaven    WI  85763           justin08@gmail.com    630-
         4       Emily Davis               038 Paige Club        East Jacob    ID  83100            eyoung@gmail.com
         5     Vanessa Moran             000 Jessica Locks       Michellefort    KS  03591        simmonsjohn@yahoo.com
         6  Sandra Patterson            0315 Collins Greens       Beverlyview    DE  70870             jack47@dixon.com  001-834
         7  Cynthia Singleton     8582 Owens Spurs Apt. 439  North Joshuamouth    TX  88781  dawnstewart@mccullough.com
         8   Kimberly Gordon         24870 Alyssa Prairie          Thomaston    ID  90540         ashley10@fischer.com     418.
        10        John Watson    283 Jonathan Ports Apt. 925         Gomezside    MA  72461          matthew30@gmail.com
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```

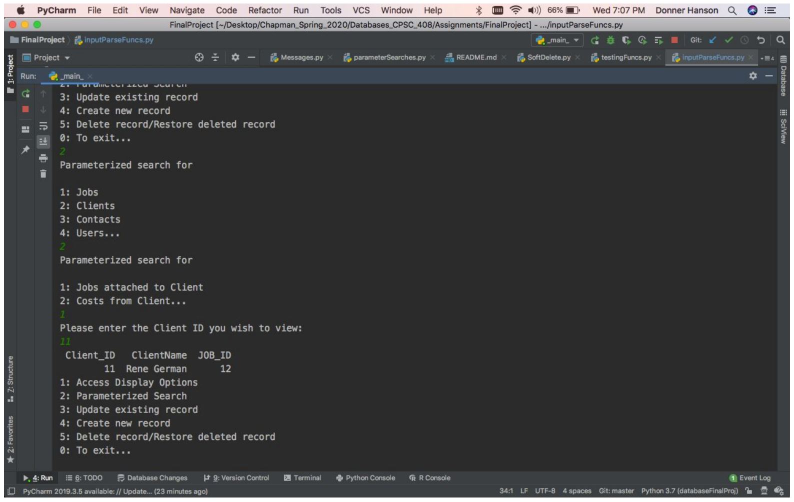

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help
FinalProject [~/Desktop/Chapman_Spring_2020/Databases_CPSC_408/Assignments/FinalProject] - .../_main_.py
updateTable.py job.py userInput.py _main_.py Messages.py parameterSearches.py README
Run: _main_
4: Run 6: TODO Database Changes 9: Version Control Terminal Python Console R Console Event Log
PyCharm 2019.3.5 available: // Update... (10 minutes ago)
238:1 LF UTF-8 4 spaces Git: master Python 3.7 (databaseFinalProj)

```
nition
tegoryTableLookup

ob cost

ntName                    Address             City State   Zip                        Email                Phone  NumJobs
Update     24415 Parker Pike Suite 440      Andersonview    IA  06537              rroy@holmes.net    143.169.3557x29119        2
elgado  57508 Sheila Mission Suite 168    Elizabethshire    WA  89402       dannybrooks@howard.com            2792524172        1
a Pham            3394 Matthew Track      Williamhaven    WI  85763           justin08@gmail.com    630-409-8062x8330        1
 Davis               038 Paige Club        East Jacob    ID  83100            eyoung@gmail.com        (448)937-1938        1
 Moran             000 Jessica Locks      Michellefort    KS  03591        simmonsjohn@yahoo.com        (756)219-4184        1
terson            0315 Collins Greens       Beverlyview    DE  70870             jack47@dixon.com  001-834-061-5340x286        1
gleton       8582 Owens Spurs Apt. 439 North Joshuamouth    TX  88781  dawnstewart@mccullough.com        (076)213-8784        1
Gordon          24870 Alyssa Prairie         Thomaston    ID  90540         ashley10@fischer.com    418.019.3045x0923        1
aynard            764 Patricia Points      Mcdanielport    ME  31135            ejacobs@yahoo.com +1-036-204-4682x66352        1
Watson     283 Jonathan Ports Apt. 925         Gomezside    MA  72461          matthew30@gmail.com        (942)498-6881        1
ns

rd

e deleted record
```

Input Checking: 

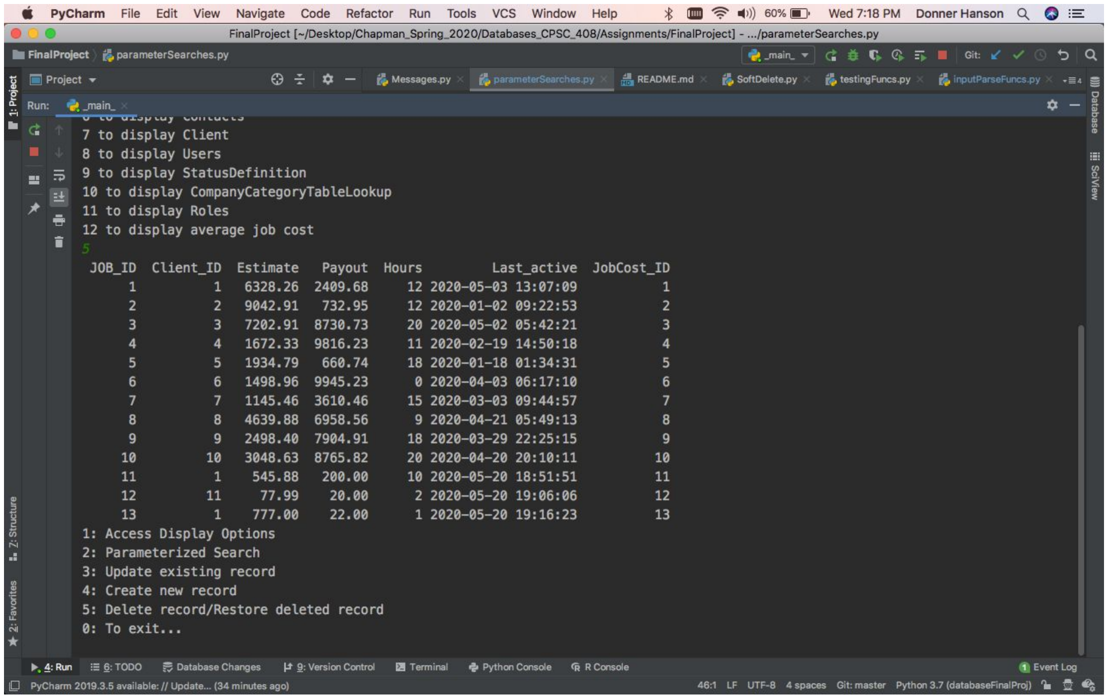
```
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
4
1: Add client and new job
2: Add user
3: Add job to existing client...
1
Enter Client's name:
Rene German
Enter Street Address:
1 University Dr.
Enter City:
Orange
Enter State (Ex: 'CA'):
P0
Enter State (Ex: 'CA'):
|
```

Client Addition Result:



```
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
2
Parameterized search for

1: Jobs
2: Clients
3: Contacts
4: Users...
2
Parameterized search for

1: Jobs attached to Client
2: Costs from Client...
1
Please enter the Client ID you wish to view:
11
 Client_ID   ClientName  JOB_ID
        11  Rene German      12
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```

Job registered with costs:



```
7 to display Client
8 to display Users
9 to display StatusDefinition
10 to display CompanyCategoryTableLookup
11 to display Roles
12 to display average job cost
5
 JOB_ID  Client_ID  Estimate   Payout  Hours         Last_active  JobCost_ID
      1          1   6328.26  2409.68     12 2020-05-03 13:07:09           1
      2          2   9042.91   732.95     12 2020-01-02 09:22:53           2
      3          3   7202.91  8730.73     20 2020-05-02 05:42:21           3
      4          4   1672.33  9816.23     11 2020-02-19 14:50:18           4
      5          5   1934.79   660.74     18 2020-01-18 01:34:31           5
      6          6   1498.96  9945.23      0 2020-04-03 06:17:10           6
      7          7   1145.46  3610.46     15 2020-03-03 09:44:57           7
      8          8   4639.88  6958.56      9 2020-04-21 05:49:13           8
      9          9   2498.40  7904.91     18 2020-03-29 22:25:15           9
     10         10   3048.63  8765.82     20 2020-04-20 20:10:11          10
     11          1    545.88   200.00     10 2020-05-20 18:51:51          11
     12         11     77.99    20.00      2 2020-05-20 19:06:06          12
     13          1    777.00    22.00      1 2020-05-20 19:16:23          13
1: Access Display Options
2: Parameterized Search
3: Update existing record
4: Create new record
5: Delete record/Restore deleted record
0: To exit...
```