Final Report

CS 419-400

Summer 2015

Group 2
Eric LaRue
Yong Lee
Tina Stahlstedt

Table of Contents

Introduction

We wrote a simple command-line / curses based controller for a SQL database. It is a step up from the command-line based tools mySQL commonly uses, in that it includes prompts to help a user create and manipulate tables and queries rather than relying on user memory to get these simple tasks done.

Our clients are any people who would like a database manager that is easier than the naked command line, but less resource intensive than the web-based options. They will act as testers, and as requesters for new services, which we would then implement.

Our first design document, taken from a web chat we had, needed to be simplified to match what we learned of Python and Curses in the last 4 weeks. We were unable to successfully configure the screen transitions we originally envisioned - our current project is much more straightforward in implementation.

Project documentation

How does your program work? With servername/address, username and password, a user first must connect to a database server. Then they can choose which database they wish to connect from a list of available databases. If a valid database is chosen, they can then proceed to add/delete/update/search tables and their data. They can also choose another database without exiting the program.

Program Structure

a) Log in screen. This is where you enter the servername, username, password.

b) Database screen. You can pick a database from a list of available databases.

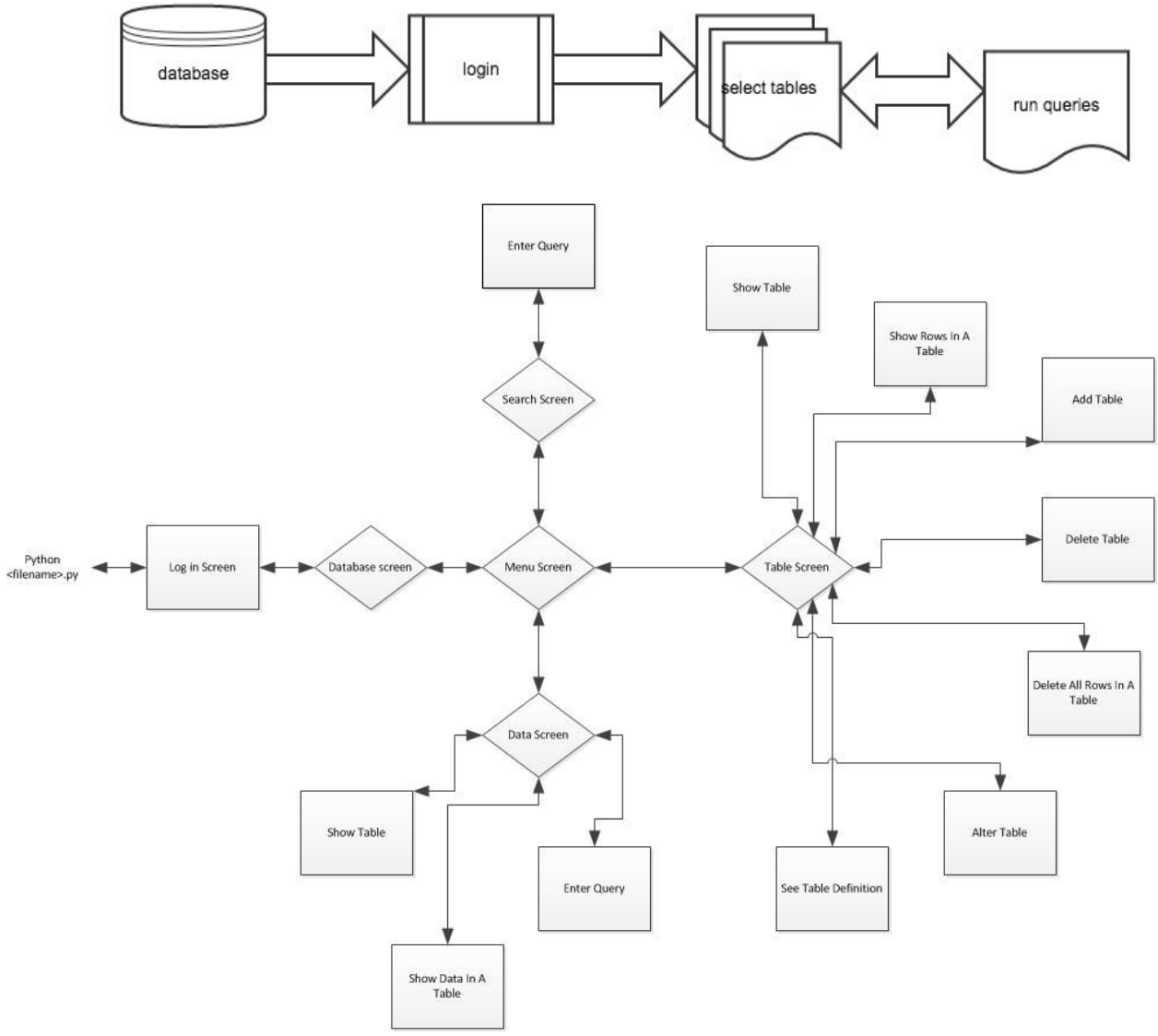
c) Menu screen. There are three options: Table, Data, Search

i) Table screen. You can add/delete/update/display the tables.

ii) Data screen. You can add/delete/update/display the data in the tables

iii) Search screen. You can enter a query and it shows the result.

Theory of Operation



Installation

No installation necessary.

Execution

Run the python file. (e.g. python <filename>.py)

System requirements

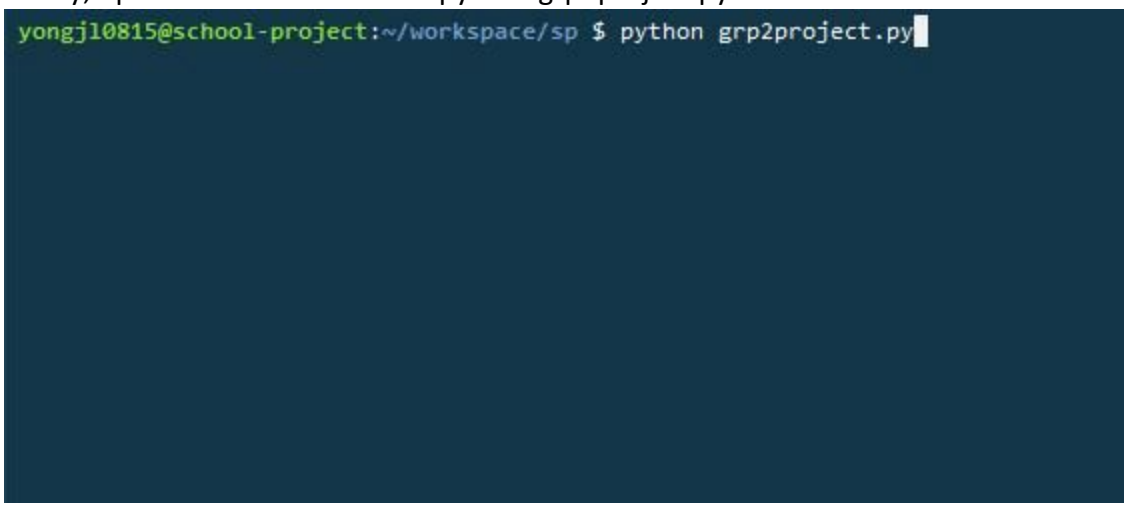
Our software is a Python 2.7x script; the only requirements are Python and mySQL. It is run normally by typing “python newmain.py” at the command line. It has been tested on Windows, Mac OSX and Linux, and runs the same on all of them.

Getting Started

To begin, have at your disposal a functional installation of MySQL, with at least one database to work with. MySQL can be found here: https://dev.mysql.com/downloads/installer/

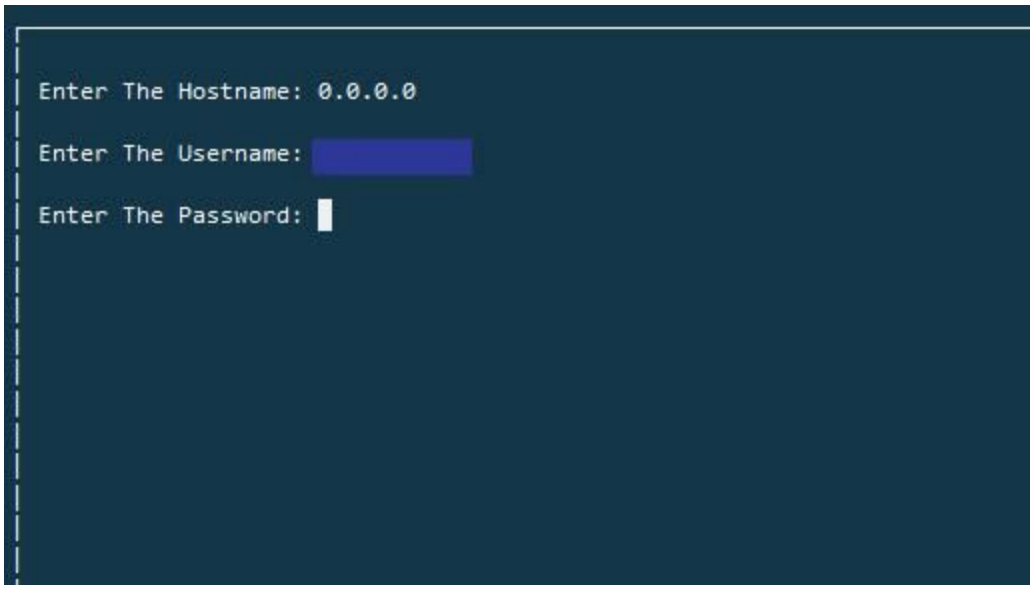
Then get Python version 2.7. The source code for most operating systems are here: https://www.python.org/downloads/release/python-2710/

Finally, open a Terminal and run “python grp2project.py”

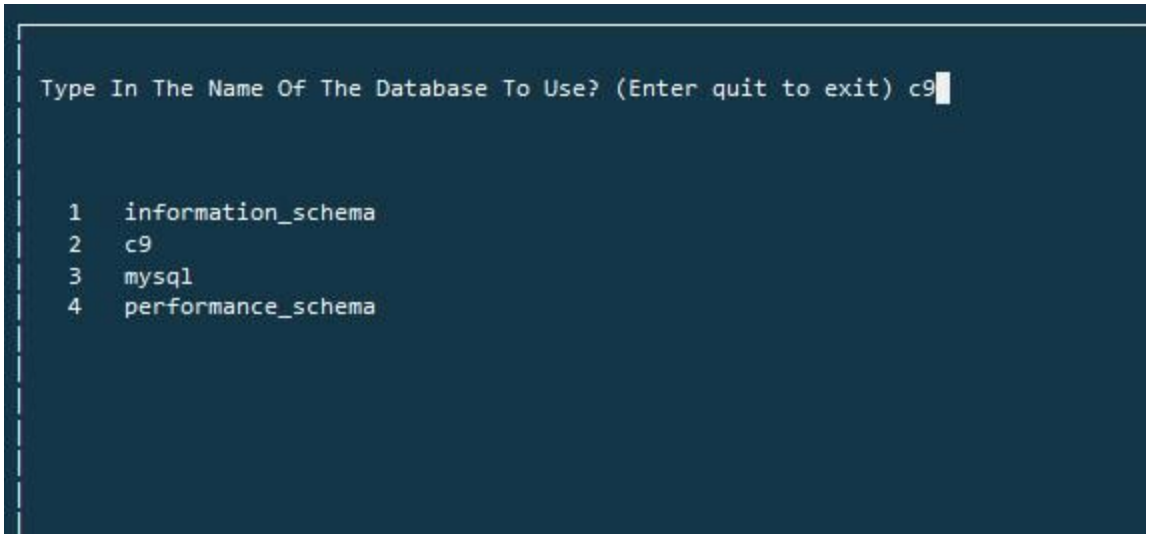


Enter in the hostname for the server your database runs on; on many local install this will be “localhost” or “0.0.0.0” or “127.0.0.1”.

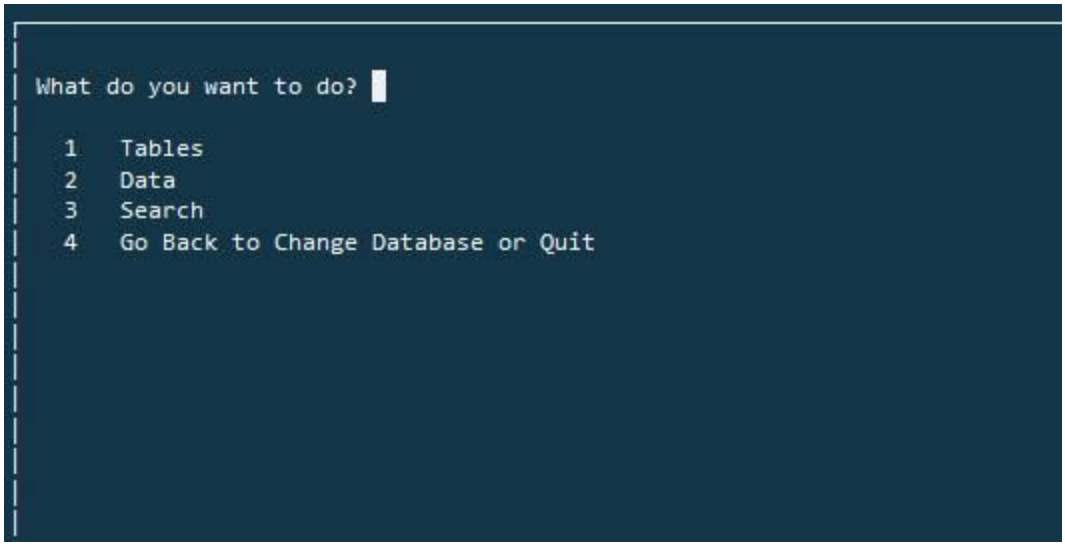
Enter the username and password that has rights to the database.



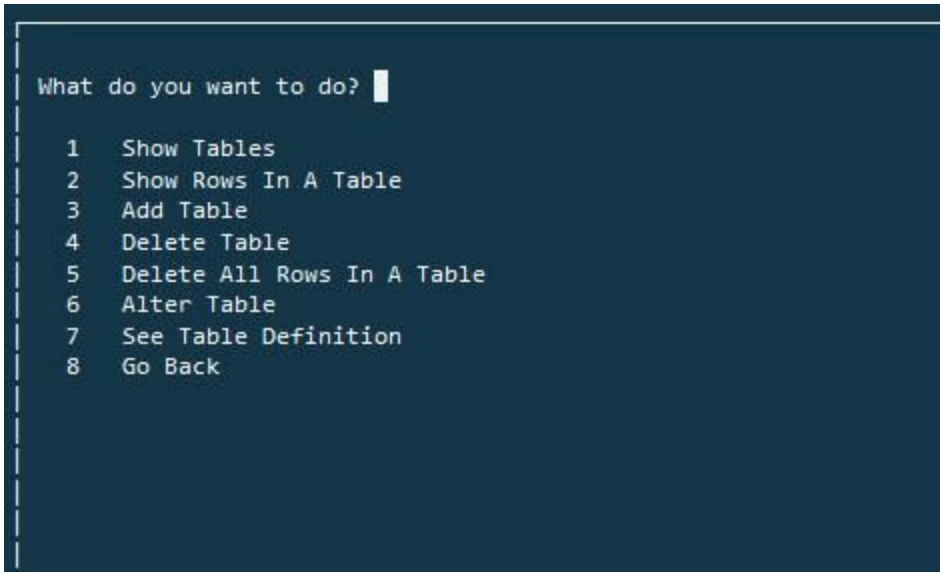
From the list provided, enter the name of the database you want to work with, eg: c9



From the choices provided, select a work option. You can use Tables (enter ‘1’) to list the tables in your database. Data (enter ‘2’) will allow you to work directly with one table. Search (enter ‘3’) will allow you to run any query on the database as a whole. To quit or pick another database, enter ‘4’.

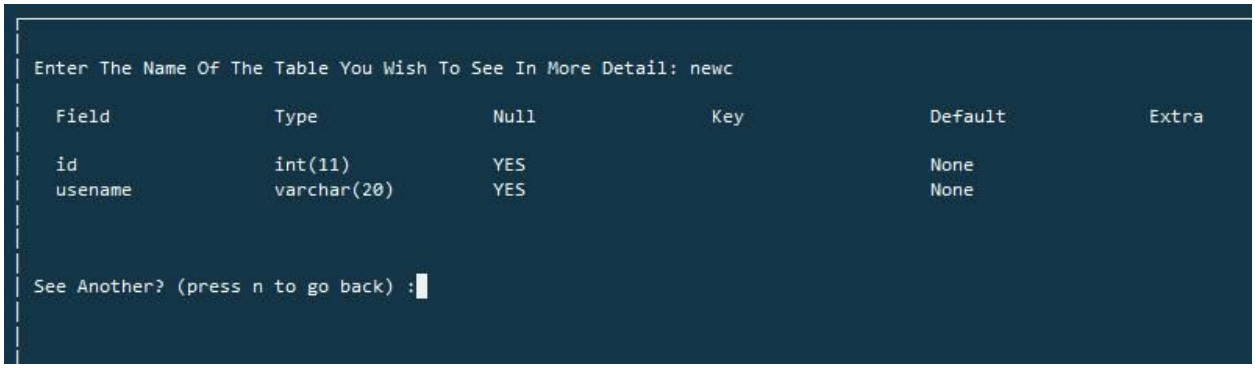


Working with Tables, you have many options:
Table Screen

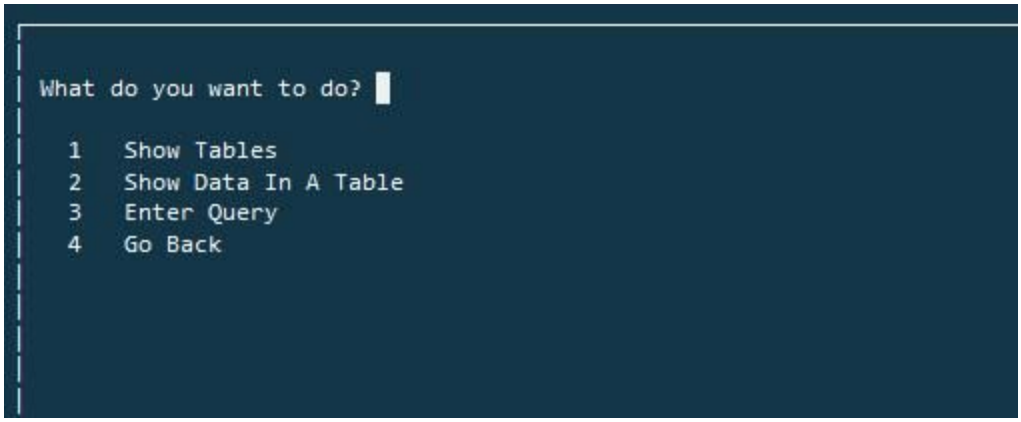


Show rows in a table

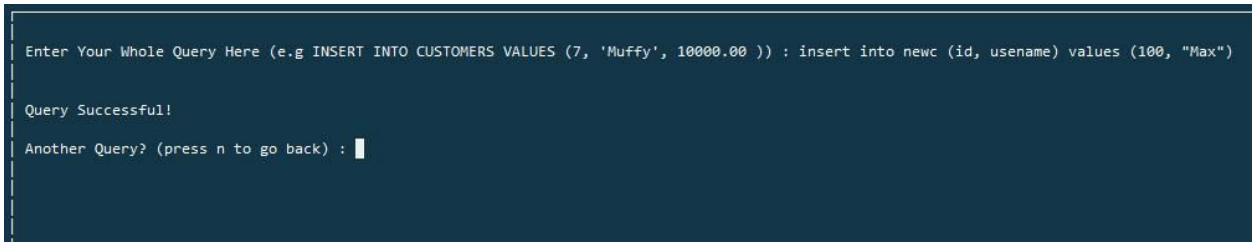
See Table Definition



From the Data screen, you have fewer, but more powerful options:
Data screen

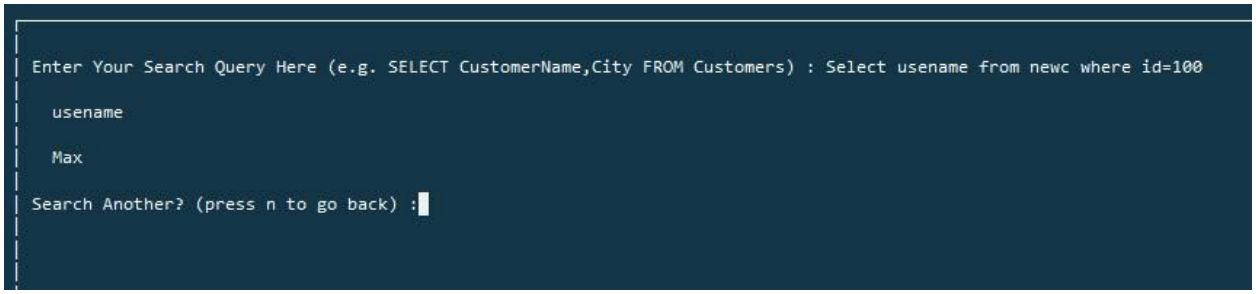


Enter Query



From the Search screen you can elect to make a SQL query:

Search Query screen



How we learned new technology

We used the following websites:

15.11. curses — Terminal handling for character-cell displays — Python 2.7.10 documentation
https://docs.python.org/2/library/curses.html

CHARMING PYTHON #6 -- Curses programming in Python: Tips for Beginners
http://gnosis.cx/publish/programming/charming_python_6.html

Code Project: Build an Ncurses UI with Python | TuxRadar Linux
http://www.tuxradar.com/content/code-project-build-ncurses-ui-python

Curses Programming with Python — Python 2.7.10 documentation
https://docs.python.org/2/howto/curses.html

MySQL Python tutorial
http://zetcode.com/db/mysqlpython/

Python MySQL Database Access
http://www.tutorialspoint.com/python/python_database_access.htm

The Curses API and Python | The Krypt
https://xerocrypt.wordpress.com/2014/09/03/the-curses-api-and-python/

Python, the Next Level (course on Udemy)
https://www.udemy.com/python-the-next-level/learn/#/

We did not find any books or people to use for this project.

What we learned

Eric LaRue

What technical information did you learn?

Since I had very little prior experience with Python, this was an opportunity for me to increase my knowledge of Python. I also learned about the ncurses architecture, which is something I had no prior knowledge of.

What non-technical information did you learn?

Mostly, this was another experience working in a remote team, collaborating over video chat (namely Google Hangouts), and sharing information over email.

What have you learned about project work?

One thing I learned is to keep better tabs on when deadlines approach. Without such awareness among all team members, often workloads can be imbalanced as one person assumes all of a task just for the sake of getting it done.

What have you learned about project management?

In this case, our team was small enough that we didn't have one particular leader - we all made ad hoc contributions towards the project design, and then coding was done as needed.

What have you learned about working in teams?

I believe this was addressed above. When working in teams, it's important for all team members to keep apprised of the schedule and plan accordingly, so that one team member doesn't take on an unfair share of the workload just to meet a deadline that caught us unawares.

If you could do it all over, what would you do differently?

See above.

Yong Lee

- What technical information did you learn? I learned a lot about ncurses. Before this course I didn't even know what ncurses was, but now I have a pretty good idea though I still have a lot to learn. (I'm fairly comfortable designing a simple CLI using ncurses now) Also this project pushed me to review some of the things I learned in other courses especially Operating Systems as I had to deal with still-learning processes and utilizing Unix commands that I haven't used in a while.

- What non-technical information did you learn? I learned a lot about virtual machines and internet based IDE that were available. Since my computer did not allow me to utilize ncurses effectively, I had to seek out other options. I looked into installing virtual machines on my laptop, using IDE online, and borrowing a computer from a friend. There were so many options for IDE and virtual machines that it was very difficult to know which one to choose, but I was able to find a nice IDE thanks to a suggestion from one of my group members and I'm looking forward to learning more about what other options are out there after this course.

- What have you learned about project work? It's not easy to go through this process when someone is not requiring you to do so. Before I had homework, readings and lectures that forced to me think about and complete each step, but without those requirements, it was kind of difficult to go through all the necessary and important steps and not just jump into coding.

- What have you learned about project management? Again without the above requirements, it was hard to determine what should be done by when and who. Managing a project so that it gets done in time and produce a good product takes a lot of communication and work. (But fun)

- What have you learned about working in teams? It's not easy working with other people especially when everyone has different schedules, but I learned to adapt to it and we embraced what worked for us. I learned that with different things going on in our lives, we cannot expect everyone to go at a same pace every week and we just have to do the best we can.

- If you could do it all over, what would you do differently? I'll be more active and initiate more communication with other members. Sometimes I didn't want to be pushy or intrusive, so I did not contact them right away when I had questions. It's something to work on.

Tina Stahlstedt

What technical information did you learn? I learned about Curses and Python programming, and how to integrate this with MySQL, which is not a thing I have done before.

What non-technical information did you learn? I learned a few more things about working with a remote team. Unlike some of my past teams, this one did not produce a strong single leader - we performed in an ad-hoc fashion, each taking lead as other schedules become complicated.

What have you learned about project work? We probably could have done better communicating with each other about the scope of the project, and where we were in it. Scheduling was especially difficult in this short summer term, since we each had many other obligations to family and work that were hard to put off.

What have you learned about project management? Given this project and the constraints on the team, it would have been good to decide on some lead responsibilities a bit earlier in the term. We each have strengths that we brought to the table, but we lost some time in waiting to see who would lead on a step.

What have you learned about working in teams? This course, and this program generally, has made me less concerned about working in teams. I used to be wary of it, now I look forward to it.

If you could do it all over, what would you do differently? See above!

Appendix 1: Essential Code Listings

How to connect to Database:

```
screen.addstr(2, 2, 'Enter The Hostname: ')
hostN = screen.getstr()

screen.addstr(4, 2, 'Enter The Username: ')
userN = screen.getstr()

screen.addstr(6, 2, 'Enter The Password: ')
passW = screen.getstr()
screen.refresh()

db = MySQLdb.connect(host=hostN, user=userN, passwd=passW)
```

How to display menu/get and process the user's input/choice.

```
def selectPage():
  c = 0
  while c != ord('4'):
        screen.clear()
        screen.border(0)
        screen.refresh()

        screen.addstr(4, 4, "1")
        screen.addstr(4, 8, "Tables")

        screen.addstr(5, 4, "2")
        screen.addstr(5, 8, "Data")

        screen.addstr(6, 4, "3")
        screen.addstr(6, 8, "Search")

        screen.addstr(7, 4, "4")
        screen.addstr(7, 8, "Go Back to Change Database or Quit")
```

```
        # draw the list on the screen
        screen.addstr(2, 2, 'What do you want to do? ')
        screen.refresh()
        c = screen.getch()

        if c == ord('1'):
        tablePage()
        screen.clear()

        if c == ord('2'):
        dataPage()
        screen.clear()

        if c == ord('3'):
        searchData()
        screen.clear()
```

How to process User's query and display result.

```
def searchData():
  c = 'y'

  while c != ord('n'):
        screen.clear()
        screen.border(0)
        screen.refresh()

        screen.addstr(2, 2, 'Enter Your Search Query Here (e.g. SELECT CustomerName,City
FROM Customers) : ')
        screen.refresh()
        x = screen.getstr()

        try:
        line = 4
        column = 4
        n = 1
```

```
cursor = db.cursor()
cursor.execute(x)
db.commit()

desc = cursor.description

w = 0

for des in desc:
screen.addstr(line, column, desc[w][0])
column += 20
w = w+1

nrow = cursor.rowcount
line += 2
column = 4

for a in range(0, nrow):
rows = cursor.fetchone()
ncol = len(rows)
for a in range(0, ncol):
screen.addstr(line, column, str(rows[a]))
column += 20
column = 4
line += 1

except:
db.rollback()
screen.addstr(5, 2, "Invalid Query!\n")

screen.addstr(line + 1, 2, 'Search Another? (press n to go back) :')
screen.refresh()
c = screen.getch()
```