To: Diana Diaz Hernendez

From: Adam Zmijewski, Jeff Hall (Full Monty)

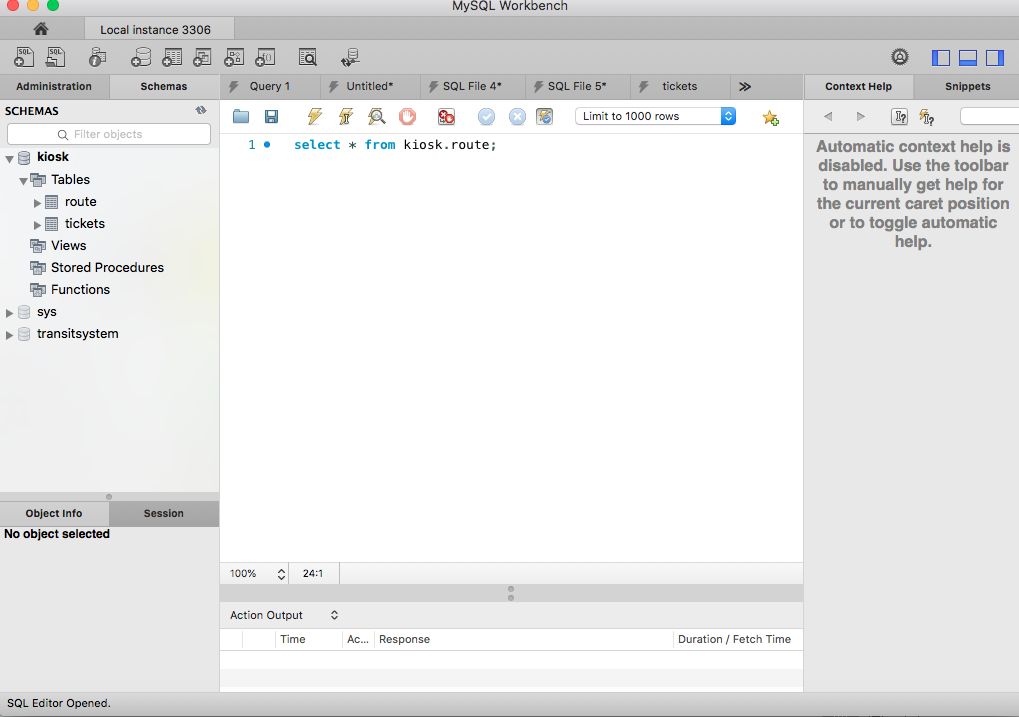
Date: April 22nd, 2019

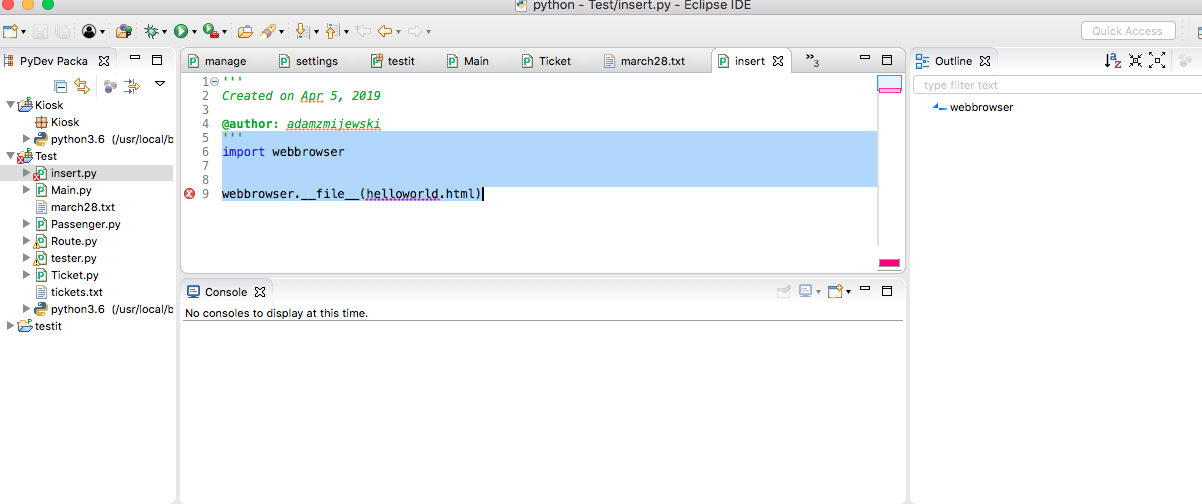
Subject: Final Project Written documentation

The written documentation for the Full Monty final project is in the sections below.

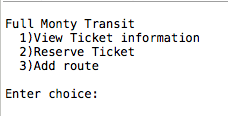
**Installation Instructions**

Firstly, there are a few things that need to be downloaded to run the project. The first thing you need is a MySQL workbench. Next, once you’ve got that downloaded you're going to need to run a query that creates a database that stores all the data of tickets into a datable. The best way to do that is to use the data export file. Just copy and paste the query code, and hit run. The database should be the same as its original. Now the database is all set up, and you should now be able to store information.



The next piece of software you need is the most recent copy of python. It very easy, and open source information to find. Once you’ve installed python, you need to install a program called eclipse ide. This program is used to run the python code, but in its default version, you can't run any python code. You're going to need to download pydev, and it's used to program python code in the eclipse ide. Once you’ve got your python programming environment all set up, make a direct text file named tickets.txt. Finally, you're going to need to install MySQL connector on the version of python you are using. The easiest way to do that is by using the command prompt.

**User Manual**



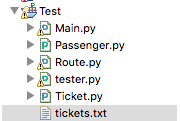
Use of the "Full Monty Transit" menu is straight-forward and easy to use. Just choose one of the three options - book a ticket for a trip; search for ticket details, or update a route in the database. For the third option, administrator login is necessary. To book a ticket, type '2' and enter the information from prompt. To search details of a ticket, type '1' and simply enter the ticket ID number. To add a route to the database, type '3', login as an administrator (username = 'root' | password = 'password'....yes, very original), and add the new route information from prompt. You will return to the options menu whenever you are finished completing your current task. You may exit the system anytime by entering 0.

**Pre Project-Team Review**

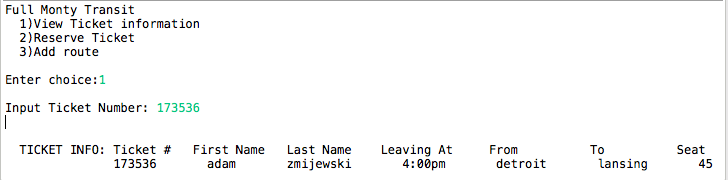
We felt we both worked well together. The GitHub experience allowed us to work together easier. We both feel it was a fun project, and it was a hands-on experience for both of us. Jeff and I thought it was a great experience working together, and we are delighted with the outcome of are the project. We feel it gave a lesson on real-world applications.

**Updated Project Proposal**

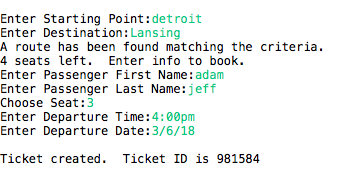
Our project proposal is pretty much the same, but it's condensed into a two-person project instead of the four people with in terms of the beginning of class. The project is a kiosk used for any mode of transportation. The program allows the user to book a ticket based on the routes in a MySQL database, insert a route, and view a ticket with the given code. Additionally, the program comes with a database used to store the information; it contains two data tables one for route and one for tickets. In our original project proposal we said it would run on an HTML page, but the program runs on console output instead.



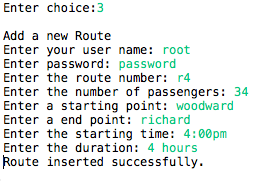
**Updated project outline**

The project is a console program that uses console display. The project has three main layers: view ticket, book a ticket and insert a route. When a user wants to view the ticket information they enter in a code they received when they first booked the ticket. When the user comes the first choice, a user gets all the necessary information to travel. The picture below shows the first selection. A user enters in a ticket number then the user views there ticket information from the ticket database.

The Reserve ticket function allows the user to book a ticket only when a starting point and destination matches a route found in the data table. The program converts every input to lowercase so that the user can type a mix of lowercase and capital letters. Once the program finds a result, the display shows how many seats are left and whether or not routes found. Next, it inserts all the information from the following inputs into the ticket data table. When the program doesn’t find a result, it doesn’t update any data tables and displays an error message. Both tables are updated when the user books a ticket. The number of tickets updates in the route data table, and then the ticket database adds another row containing the ticket information.



Finally, it allows you to insert a route only with secure authentication. A user must type in ‘root’ for their username, and ‘password’ for the password. When the authentication is complete, a user can insert a new route based on the console inputs.



**Final architecture documentation**

Jeff Hall

au8558

CSC 4992

Final Project

O/S - Windows 10

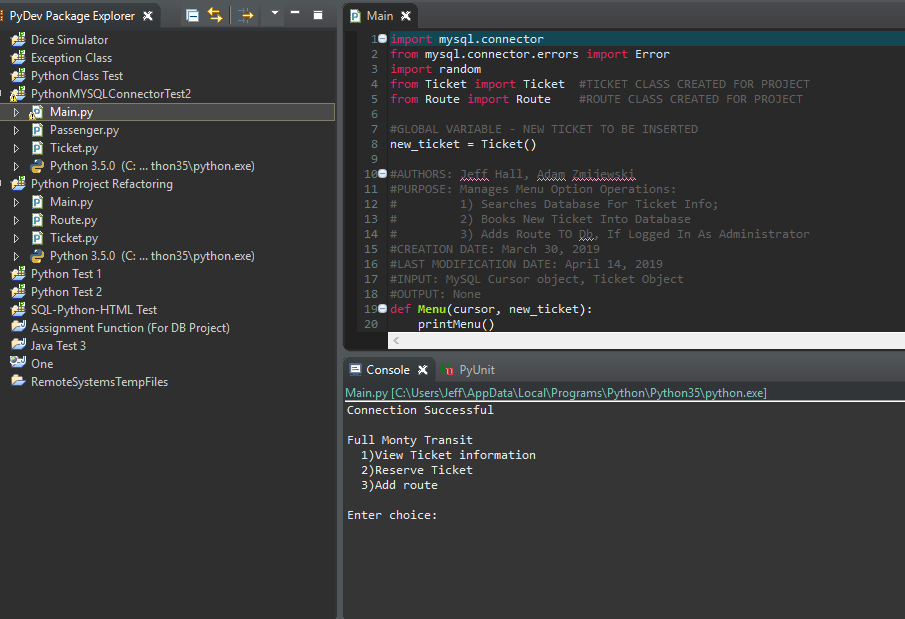
System Type - 64-bit Operating System, x64-based Processor

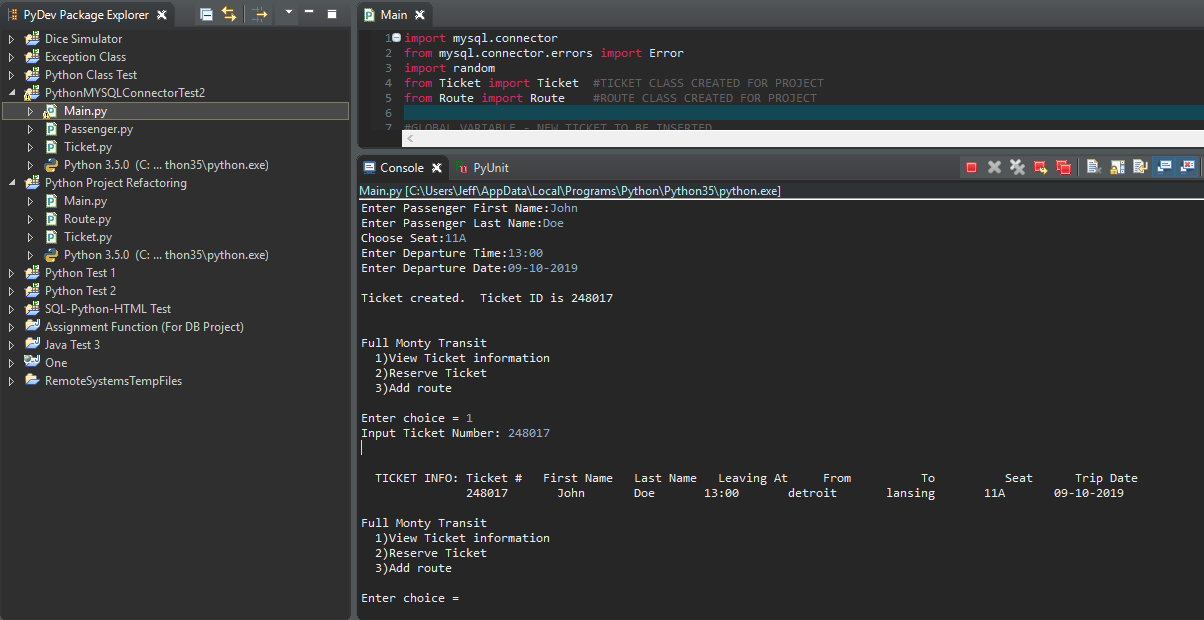
Processor - Intell Core i3-4005U CPU @ 1.70HGz

Installed RAM - 4.00 GB

MAC Address - 4C-BB-58-E4-35-C2

Running the kiosk program





Adam Zmijewski

Fh7020

CSC 4992

Final Project

O/S – Mac OS High Sierra

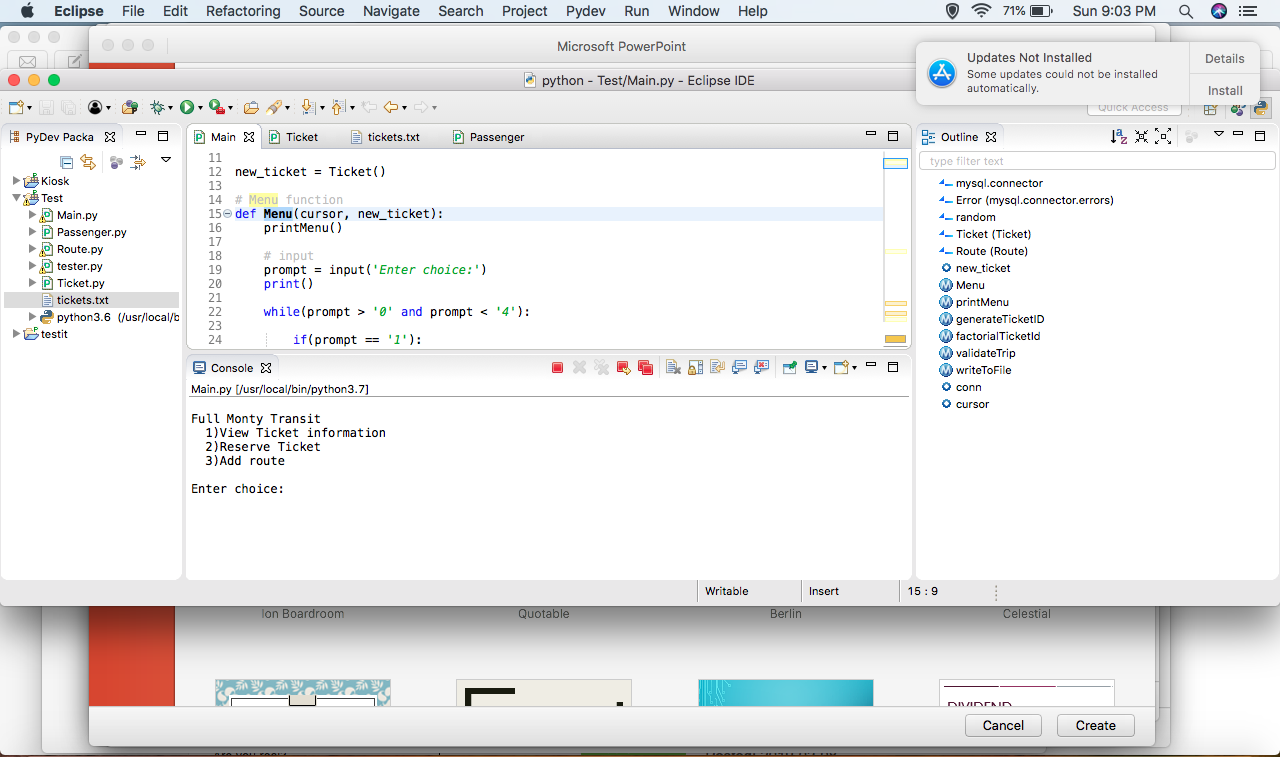
System Type - 64-bit Operating System

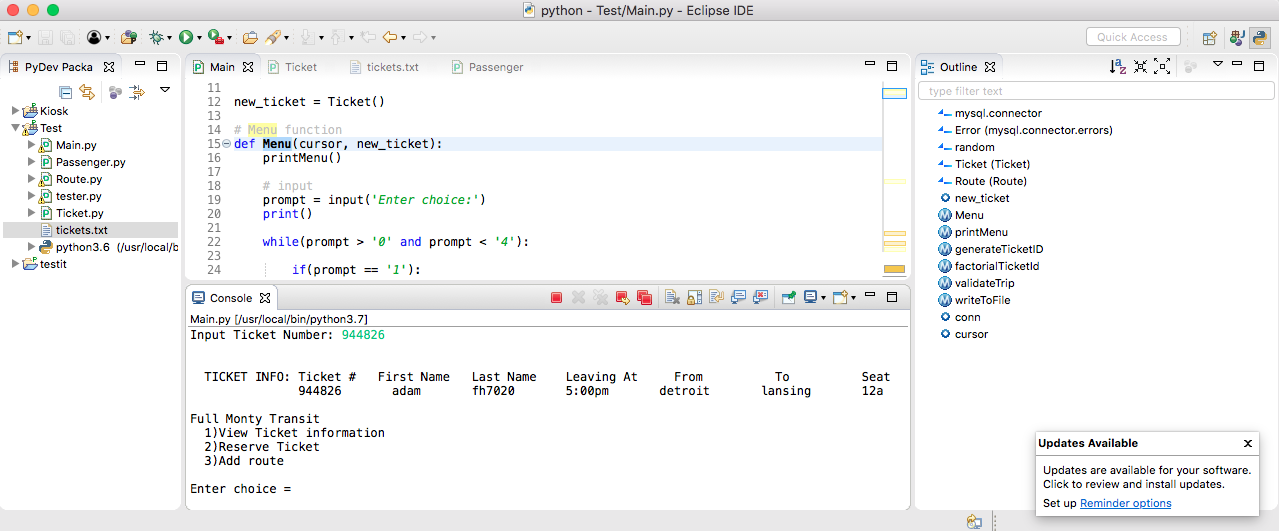
Processor - 2.4 GHz Intel Core 2 Duo

Installed RAM - 8.00 GB

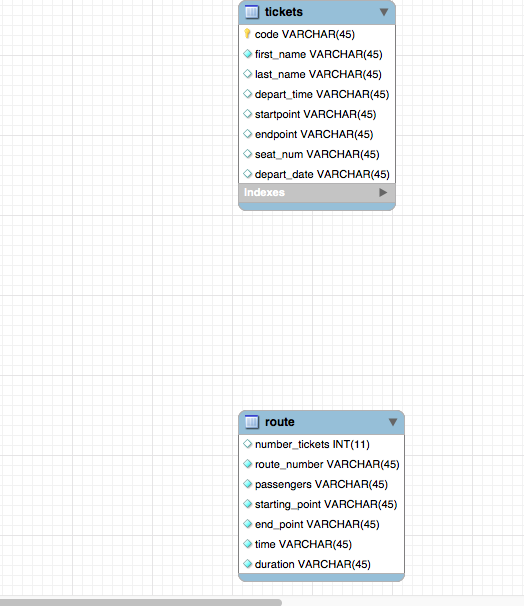
MAC Address - d8:a2:5e:8f:67:85

Running the Kiosk Program





The picture below summarizes all the data fields in each database for the project.

****

(Download Links)

<https://www.eclipse.org/downloads/>

<https://www.python.org/downloads/>

<https://dev.mysql.com/downloads/workbench/>

<https://dev.mysql.com/downloads/connector/python/>