

State University of New York at New Paltz

Kaitlin Hoffmann

FINAL PROJECT REPORT

Team ID: 12
Project Type: Clothes
CS Shared Folder: b-f19-06
E-Business, Professor Pham, Fall 2019

URL (Web Address):
<https://cs.newpaltz.edu/e/b-f19-06/>

Name	PROGRAM					DATA			
	Catalog	Checkout	B2Bserver	B2Bclient	Files for B2B	Users/ Customers	Inventory	Orders	
Form	n/a	- checkout.cgi	- newServer.cgi - Server.java - Server.class - InServer.class - OutServer.class	- pay.cgi - Client.java - Client.class - As.dat - Qs.dat - f-As.dat - f-Qs.dat	- necPorts.txt - customer_t - cart_t	- inventory_t	-	-order_t - BankRequest_t - ShippingRequest_t - ITRequest_t - TaxRequest_t	
Processing (CGI)	- catalog.cgi - view_item.cgi	- checkout.cgi - confirmCheck.cgi - pay.cgi							

Table of Contents

1. Business Model	Page 4
1.1 Business Context and Goals	Page 4
1.2 Technical Requirements and Technologies	Page 4-5
2. Architecture	Page 5
2.1 Website	Page 5
2.2 API	Page 6
3. Protocols	Page 7
3.1 Descriptions	Page 7
3.2 Diagrams	Pages 8-9
4. Design	Page 9
4.1 Databases	Pages 9-12
4.2 Text Files	Pages 12-13
5. Implementation	Page 14
5.1 Website Address of The Kitty Cabin	Page 14
5.2 Maps of Subdirectories and Files and Descriptions	Pages 14-15
5.3 B2B	Page 16-17
6. Website Demo/Test	Page 18
6.1 Registering - Transaction 0: Individual Account	Pages 18-20
6.2 Logging in - Transaction 1	Page 20
6.3 Viewing Items - Transaction 2	Pages 21-23
6.4 Adding to Cart - Transaction 3	Pages 23-24
6.5 Removing items from Cart - Transaction 3	Pages 24-25
6.6 Checking Out - Transaction 4	Pages 25-27
6.7 Confirm Order - Transaction 5	Pages 27-29
7. B2B Demo/Test	Page 29
7.1 Registering: Business Account	Pages 29-31
7.2 Logging In	Pages 31-34
7.3 Order Through Staff B2B	Pages 35-39
7.4 Inside StaffB2B CGI program	Page 39-40
7.5 Inside Ac File	Page 40
8. Developer Guide	Page 41
8.1 How to Run/Setup The Kitty Cabin	Page 41
8.2 Issues to Look Out For	Page 41
8.3 Changes Needed When Moving Folders	Page 41

9. Challenges	Page 42
9.1 Errors Encountered and How they were Fixed	Page 42
9.2 Other Challenges	Page 42
10. Conclusion	Page 43
11. References/Appendixes	Page 43-44

1. Business Model

1.1 Business Context and Goal

What kind of business are we?

A retail e-business meant for cat lovers in need of some comfortable clothes who want to have the cutest cats on their clothes...my cats!

Target Demographic:

Adults and children, men and women, anyone is a potential customer! Unless you hate cats. We don't want your service. There were over 61 million households in 2017-2018 that owned a cat in the United States according to a survey from the American Veterinary Medical Association. All potential cat lovers and customers!!

Catalog Items:

Sweat Pants: Available in either shorts or long pants. Bottoms are layered images of Gigi and her cute face.

White T-Shirt: Shirts with a picture of Harry in his coolest pose.

Black T-Shirt: Shirts with a picture of Harry in an even cooler pose.

Tank Tops: Tanks with an image of Earl hanging out enjoying life.

Socks: Socks with a headshot of Jerry and his blank expression.

1.2 Technical Requirements and Technologies

Platform and Specifications:

Apache server with Linux OS

We're expecting a few hundred customers daily so At least 16 GB of RAM with 2 intel core CPUs (2.66 GHz).

Middleware:

Keeping it simple! CGI. Not intending to use a framework.

Programming Languages:

Web Pages: HTML, CSS, JavaScript

CGI: Python

SOCKETS: Java

Database: MySQL

The Kitty Cabin e-business system will have a Website and a socket based API:

A. Website:

Users: Admin, Public and Registered Customers

Components: Home Page, Registration, Login, View Catalog, Shop (add/delete items from cart), Checkout, Order and Pay for items.

B.API:

Users: Retail and service businesses such as Shipping, Bank, Mayor and IT Services businesses.

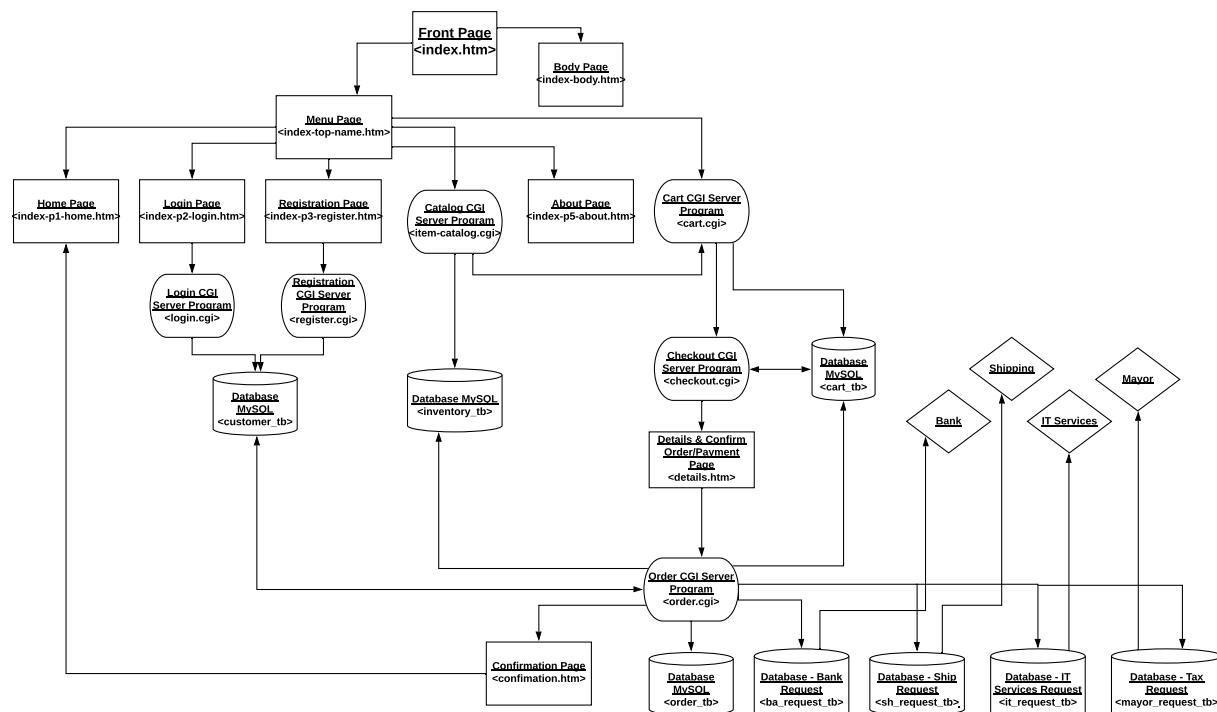
Components:

Client Socket that connects to a server socket to ask a question. Allows The Kitty Cabin to connect to the Shipping, Bank, Mayor and IT Services businesses in order to use their services.

Server Socket that connects to client socket to give a response. Allows other businesses to order from The Kitty Cabin.

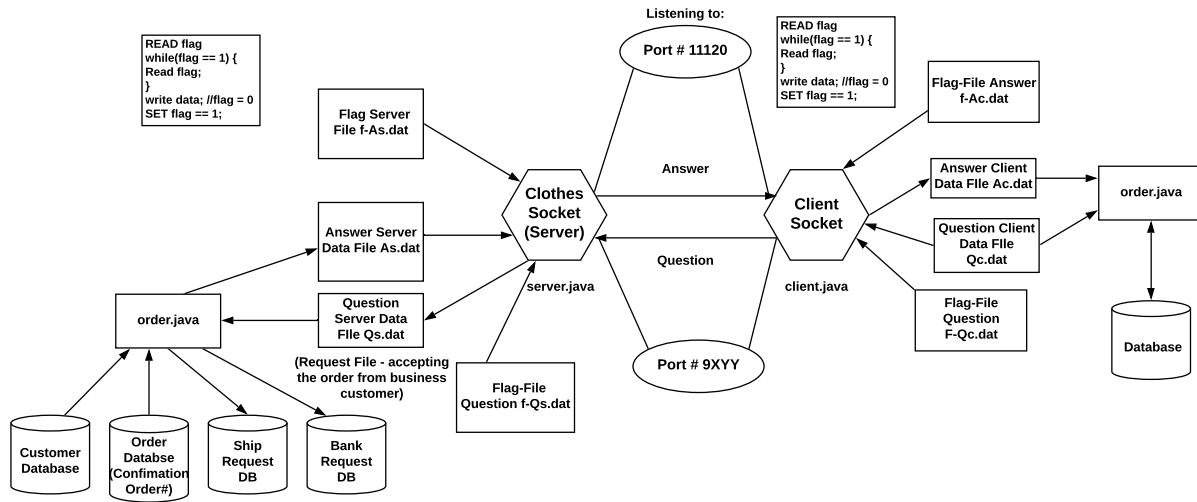
2. Architecture

2.1 Website



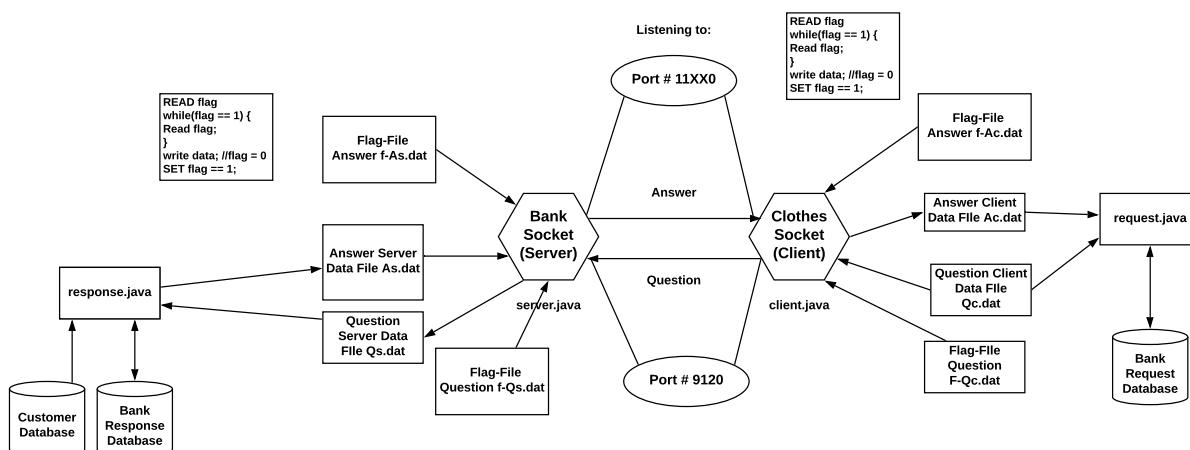
2.2 API

THE KITTY CABIN AS THE SERVER



THE KITTY CABIN AS THE CLIENT

3.



***API Architecture for Shipping, IT Services and Major looks the same with different Response and Request Database.**

Protocols

3.1 Descriptions

T0: Registration

Allows customers to create an account in order to shop and buy items. Customer must fill form out with name, password, address, debit card number.

T1: Login

Allows customers to log into their account allowing them to shop and purchase items. Must fill out form with name and password. Uses Customer database.

T2: View Catalog

Customers can look at item pages that are created through catalog CGI program. Uses Inventory Database.

T3: Shop (add/delete)

Customers can add and delete items from their cart. Uses a cart CGI program. Uses inventory and cart database.

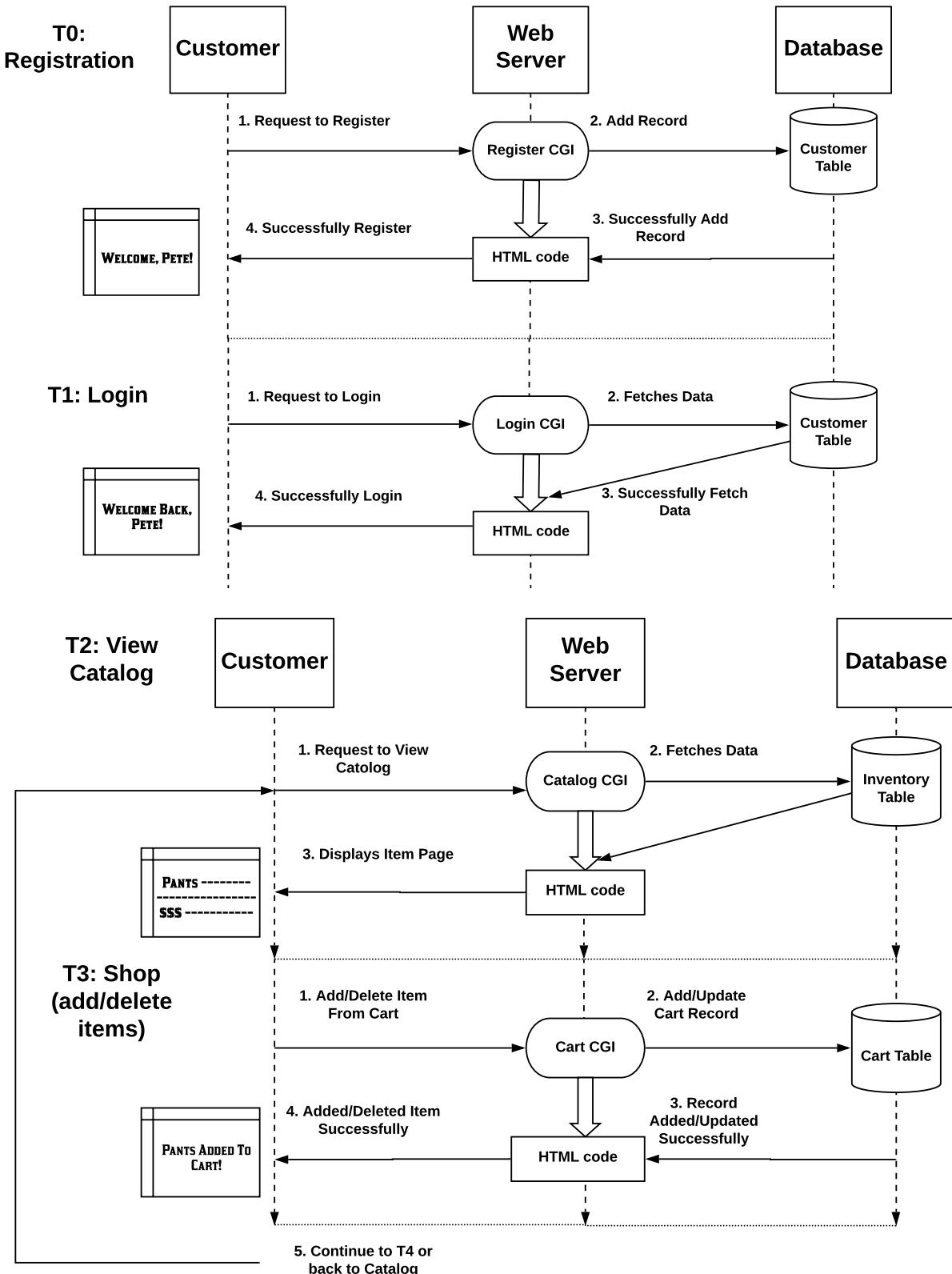
T4: Payment/Checkout

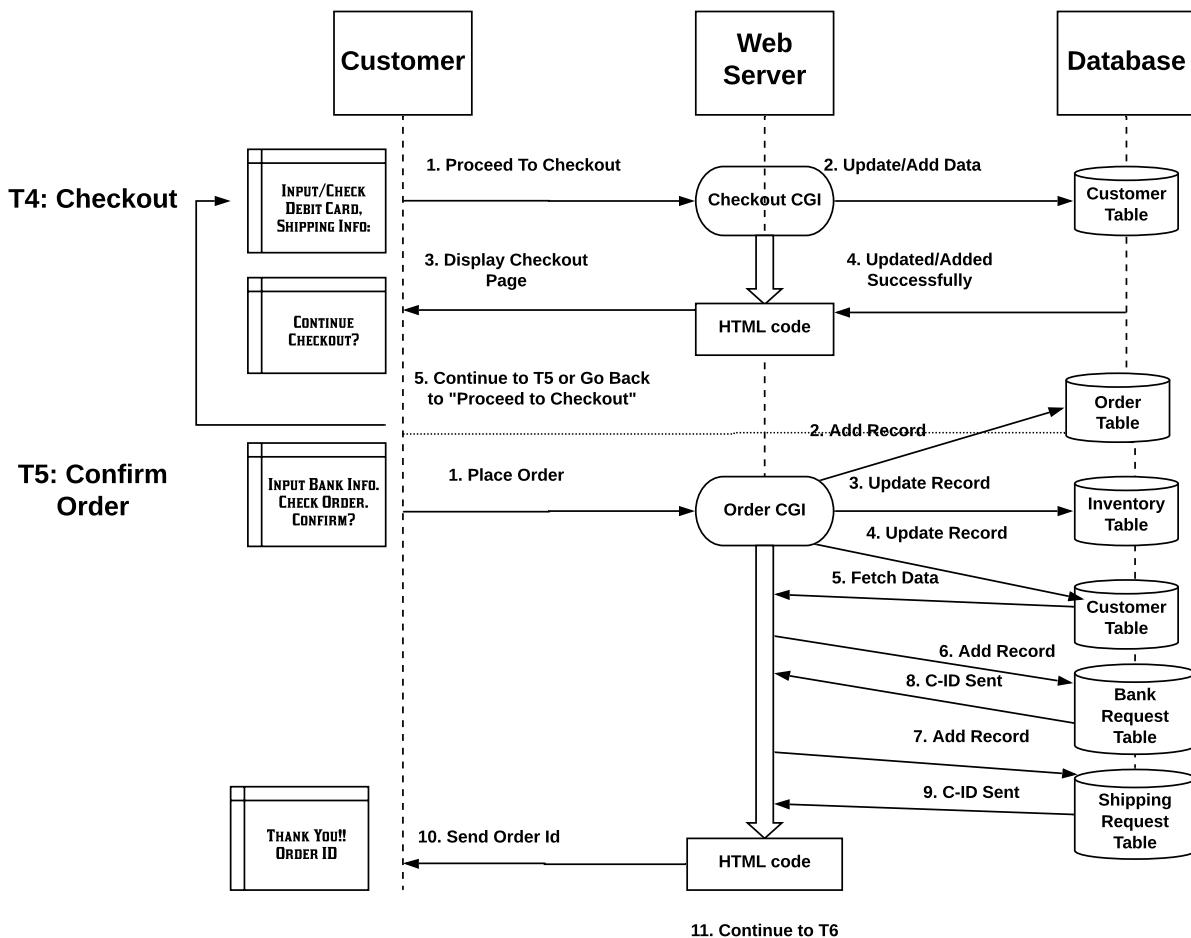
Customers provide and/or confirm debit card information through a checkout form created from a checkout CGI program. Uses customer database.

T5: Confirm Order

Allows customers to confirm orders and input bank information. Order CGI program updates inventory, cart, order and customer database. Fetches data from customer table. Ship and bank requests are sent through sockets. Shipping and banking companies sends confirmation number requests which are sent through sockets. Shipping and banking companies sends confirmation number allowing payment to be confirmed. Customer is sent back to the homepage.

3.2Diagrams





4.Design

4.1 Databases

```
MariaDB [b_f19_06_db]> show tables;
+-----+
| Tables_in_b_f19_06_db |
+-----+
| BankRequest_t
| ITRequest_t
| ShippingRequest_t
| TaxRequest_t
| cart
| customer_t
| inventory_t
| order_t
+-----+
8 rows in set (0.04 sec)
```

Snippets of data from database tables:

Customer Record - customer_t;

```
MariaDB [b_f19_06_db]> select * from customer_t;
```

customer_id	password	email	customer_name	customer_type	customer_address	debit_card_num
120001	ggg	kaitlinhoffmann@gmail.com	Kaitlin C Hoffmann	Individual	1 Cherry Ave	12345
120002	12549	jerry@gmail.com	jerry	Individual	2554 State Route	167725
120003	12345	pete@gmail.com	Pete Jogs	Individual	4 Leaf Drive	1234562
120004	ggg	millo200@yahoo.com	Herminio Vazquez	Individual	120 West 91st Street Apt 2J	12345

Inventory Record - inventory_t;

```
MariaDB [b_f19_06_db]> select * from inventory_t;
```

item_id	current_quantity	current_price	size	total_in	total_sold	item_name	item_picture	item_descrip
121000	49935	19.95	One Size Fits All	49935	108	Harry's Cool Shirt	http://cs.newpaltz.edu/e/b-f19-06/HarryShirt.jpeg	NULL
121001	49958	19.95	One Size Fits All	49958	42	Harry's Cooler Shirt	http://cs.newpaltz.edu/e/b-f19-06/harryshirt2.jpeg	NULL
121002	49948	3.95	One Size Fits All	49948	52	Jerry's Sox	http://cs.newpaltz.edu/e/b-f19-06/jerrysock.jpeg	NULL
121003	49948	25.95	One Size Fits All	49948	52	Gigi's Bottoms	http://cs.newpaltz.edu/e/b-f19-06/gigipants.jpeg	NULL
121004	49963	11.95	One Size Fits All	49963	37	Earl's Tank	http://cs.newpaltz.edu/e/b-f19-06/earltank.jpeg	NULL

Order Record - order_t;

```
MariaDB [b_f19_06_db]> select * from order_t where order_conf is not null;
```

order_id	customer_id	item_id_ordered	quantity	ship_method	ship_address	debit_card_num	order_conf
122482	120002	121000	1	1	2554 State Route	167725	1200122482
122488	120040	121004	2	1	vvvv	1234	1200122488
122490	120002	121000	1	1	2554 State Route	167725	1200122490
122492	120002	121004	2	1	2554 State Route	167725	1200122492
122493	120002	121000	2	1	2554 State Route	167725	1200122493

Cart Record - cart_t; (rows get deleted after a successful checkout)

```
MariaDB [b_f19_06_db]> select * from cart;
+-----+-----+-----+-----+
| cart_id | customer_id | item_id | Quantity |
+-----+-----+-----+-----+
| 123185 | 120019 | 121001 | 1 |
| 123195 | 120022 | 121002 | 1 |
| 123219 | 120056 | 121000 | 1 |
| 123221 | 120048 | 121004 | 1 |
+-----+-----+-----+-----+
4 rows in set (0.07 sec)
```

Bank Request Record - BankRequest_t; (I apologize for format, not sure how to fix)

```
MariaDB [b_f19_06_db]> select * from BankRequest_t;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | PayeeAccount | PayorAccount | Amount |
+-----+-----+-----+-----+-----+-----+
| 120002 | 12549 | 1191234567 |
| 888888 | 4676 | 19.95 |
| 120002 | 12549 | 1191234567 |
| 888888 | 123123 | 25.95 |
| 120002 | 12549 | 1191234567 |
| 888888 | 12133 | 11.95 |
| 120002 | 12549 | 1191234567 |
| 888888 | 4676 | 19.95 |
+-----+-----+-----+-----+-----+-----+
```

Shipping Request Record - ShippingRequest_t;

```
[MariaDB [b_f19_06_db]> select * from ShippingRequest_t where CRequestID is not null;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | ItemIDOrdered | ShipMethod | ShipAddress |
+-----+-----+-----+-----+-----+-----+
| 120002 | 12549 | 2191234567 | 121004 | 1 | 2554 State Route |
| 120002 | 12549 | 2191234567 | 121000 | 1 | 2554 State Route |
| 120002 | 12549 | 2191234567 | 121004 | 1 | 2554 State Route |
+-----+-----+-----+-----+-----+-----+
```

IT Services Request Record - ITRequest_t;

```
MariaDB [b_f19_06_db]> select * from ITRequest_t;
```

CRequestID	ItemId	Quantity	SaleAmount	BankAccount	ShipMethod	ShipAddress	ITAccount	ITPassword
4191234567	121000	1	20.00	4676	1	2554 State Route	1234567	abcd
4191234567	121004	1	12.00	2424	1	2554 State Route	1234567	abcd
4191234567	121003	1	50.00	12133	1	2554 State Route	1234567	abcd
4191234567	121002	1	50.00	12133	1	2554 State Route	1234567	abcd

Mayor/Tax Request Record - TaxRequest_t;

```
MariaDB [b_f19_06_db]> select * from TaxRequest_t;
```

CRequestID	Amount	BankAccount	TaxAccount	TaxPassword
3191234567	12.00	2424	7654321	dcba
3191234567	12.00	12133	7654321	dcba
3191234567	26.00	12133	7654321	dcba

4.2 Text Files

Java Client Program and Changing ports and Files with necPorts.txt -

In order to change the ports, the flag files, and question and answer files, inside the Client.java program, I made it so the instance variables/attributes are set by reading file necPorts.txt:

```
public Client (String host) throws IOException {  
  
    Scanner sc = new Scanner(new File("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/necPorts.txt"));  
    Port_to_Server = sc.nextInt();  
    Port_from_Server = sc.nextInt();  
  
    QflagName = sc.next();  
    QnameAtC = sc.next();  
    QnameAtS = sc.next();  
  
    AflagName = sc.next();  
    AnameAtC = sc.next();  
  
    AnameAtS = sc.next();
```

Inside the pay.cgi, b2b.cgi and newSerb.cgi program where I am calling my Client Server, I change the contents of the necPorts.txt by writing over it every time I need to call a different Server. Below is an example altering necPorts.txt and calling the Test Port for IT Services inside my pay.cgi program:

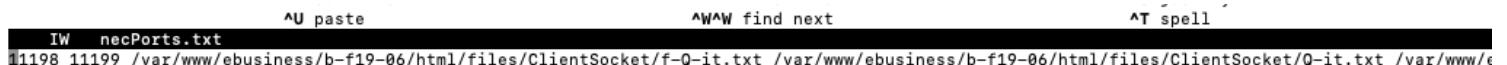
```
itInfo = str(item) + " " + str(quantity) + " " + str(totPrice) + " " + str(bankN) + " " + shipMethod + " " + sAdd + " " + str(itAcc) + " " + itPass
itFile = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-it.txt", "w")
itFile.write(itInfo)
itFile.close()

#changing port:
changePort = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/necPorts.txt", "w")
changePort.write("11198 11199 /var/www/ebusiness/b-f19-06/html/files/ClientSocket/f-Q-it.txt /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-it.txt /var/www/ebus
changePort.close()

#calling Client
p1 = subprocess.Popen(["/usr/bin/java", "Client"], stdout=subprocess.PIPE)
print p1.stdout.read()

with open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-it.txt", "r") as file:
    itRes=file.readline() #file.read()
    itConf=file.readline()
    itConf = str(itConf)
    itRes=map(str,itRes)
    itRes2=itRes[0]
```

Below is the new contents of necPorts.txt now:



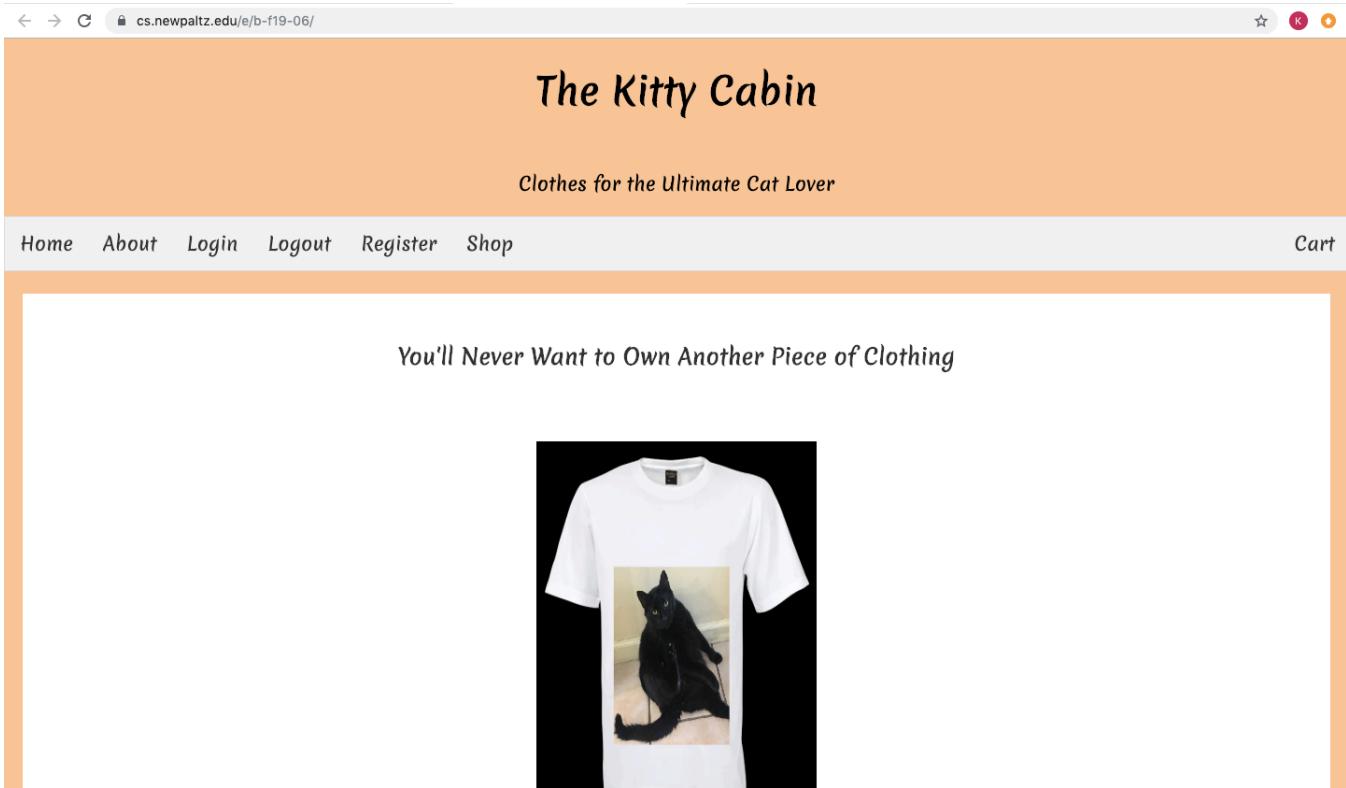
The screenshot shows a terminal window with the following text in the buffer:

```
11198 11199 /var/www/ebusiness/b-f19-06/html/files/ClientSocket/f-Q-it.txt /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-it.txt /var/www/ebus
```

5. Implementation

5.1 Website Address of The Kitty Cabin

<https://cs.newpaltz.edu/e/b-f19-06/>



5.2 Maps of Subdirectories and Files and Descriptions

Permissions:

Due to how the shared folder was set up, permission codes did not need to be changed. However, if this were a different system, all CGI programs would need their permissions to change to 755. (chmod program.cgi 755)

Programming Languages:

All CGI programs are written in python.

HTML files are written in html and css.

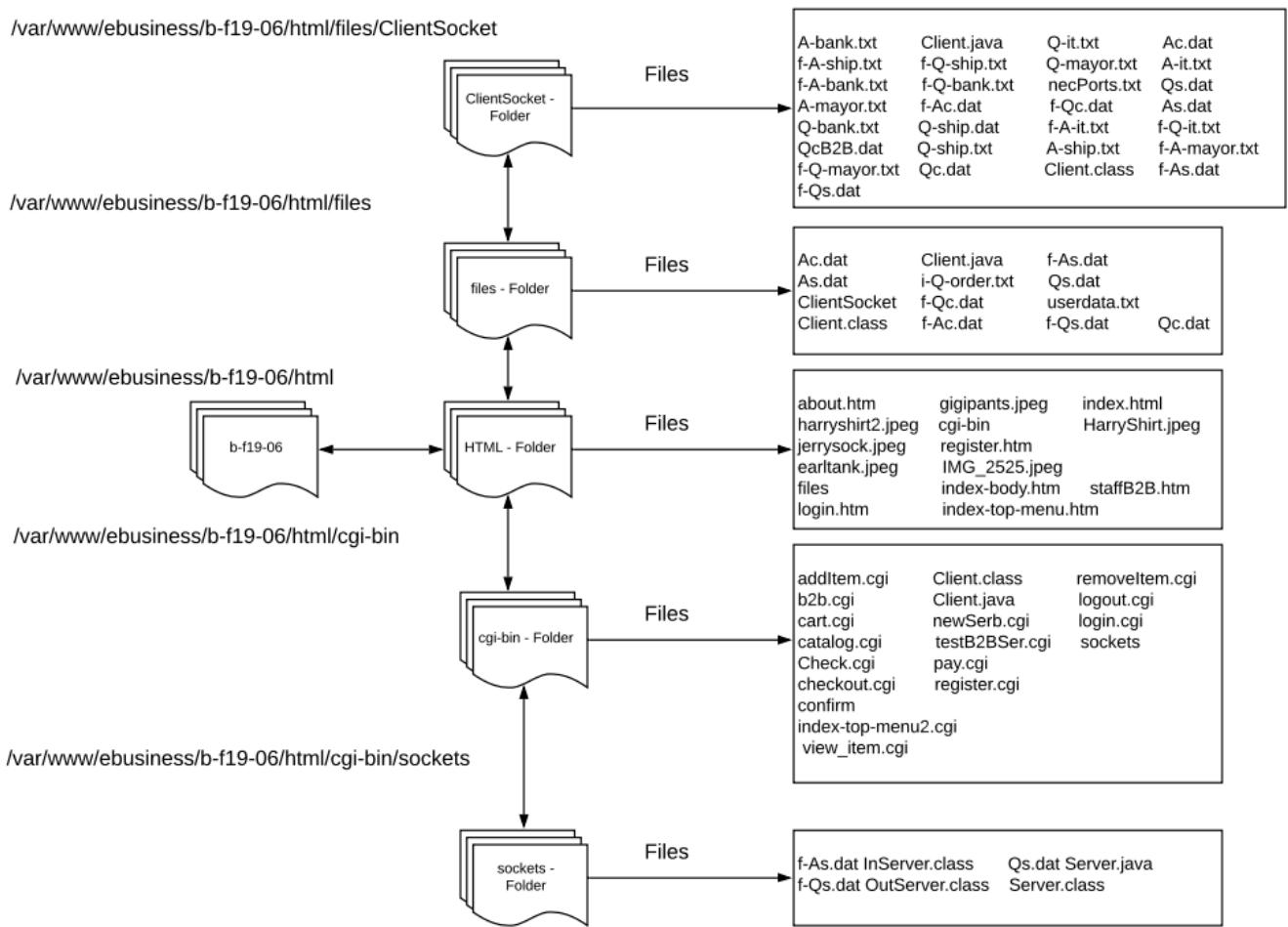
.java and .class files are written in Java.

necPorts.txt and Qs.dat files are written on one line with spaces in between each item.

As.dat files have a reply number on first line and a confirmation number on second line if reply number is 0.

.jpeg files were transferred over with CyberDuck.

Map Diagram:



5.3 B2B

B2B Server - Done:

Port numbers are defined in the Server.java program (under /var/www/ebusiness/b-f19-06/html/cgi-bin/sockets). The Kitty Cabin **Data Port** is **11120**. The **Reply Port** is **11121**. The port numbers are hardcoded inside the Server.java program (**lines 15 and 16**). The Server.java program is provided by courtesy of Dr. Hanh Pham. To make the B2B server program work so other companies can order from The Kitty Cabin through their Client program, I created a python program, **newSerb.cgi** (**under /var/www/ebusiness/b-f19-06/html/cgi-bin**), which is called inside the runnable **InServer** class in the Server.java program:

```
try {
    Process p = Runtime.getRuntime().exec("python /var/www/ebusiness/b-f19-06/html/cgi-bin/newSerb.cgi"); //serb.cgi
}
catch (Exception e) {
    System.out.println(e);
}
```

Inside newSerb.cgi, companies must send over their customer id number, password, the item id they want to purchase, the quantity they want purchase, and shipping method:

```
##read the file
with open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Qs.dat", "r") as file:
    custQues=file.read()
    custQues=str(custQues)
##iterate through the words in the string
cusQuest = custQues.split(" ")
custId = cusQuest[0]
custPass = cusQuest[1]
itemId = cusQuest[2]
quant = cusQuest[3]
sMethod = cusQuest[4]
[db = MySQLdb.connect("localhost", "b_f19_06", "v5qteo", "b_f19_06_db")
myCursor = db.cursor()

sql = "SELECT * FROM customer_t WHERE customer_id = '%s'"%(custId)
answer = "0"

try:
    myCursor.execute(sql)
    output = myCursor.fetchall()
    for row in output:
        custId = row[0]
        custPass = row[1]
        custType = row[4]
        sAdd = row[5]
        dCard = row[6] #This will be bank account
except:
    answer = "1"
    print "Account does not exist"

if (custType != "Business"):
    answer = "2"
    print "Wrong account type, try again with a BUSINESS account."
```

After this information is verified through The Kitty Cabin databases, The Kitty Cabin **Client Server** is called.

B2B Client - Done:

The **Client Server** is called Client.java under **/var/www/ebusiness/b-f19-06/html/cgi-bin** and is also provided by courtesy of Dr. Hanh Pham. Any other Client Server programs seen anywhere else in my shared folder is strictly used for my own personal testing purposes. I call/use The Kitty Cabin Client Server in **pay.cgi**, **b2b.cgi**, and **newSerb.cgi** which are all under cgi-bin alongside Client.java. **Port numbers** and file names are changed inside each of these programs using **necPorts.txt** under **/var/www/ebusiness/b-f19-06/html/files/ClientSocket/**. I wrote python programs to overwrite necPorts.txt the new port numbers and files used. Example:

```
## Connecting to Shipping Company:  
  
shipInfo = str(custId) + " " + str(custPass) + " " + str(itemId) + " " + str(sMethod) + " " + str(sAdd)  
shipFile = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-ship.dat", "w")  
shipFile.write(shipInfo)  
shipFile.close()  
  
#changing port:  
changePort = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/necPorts.txt", "w")  
changePort.write("11194 11195 /var/www/ebusiness/b-f19-06/html/files/ClientSocket/f-Q-ship.txt /var/www/ebusi  
changePort.close()  
  
#calling Client  
p1 = subprocess.Popen(["/usr/bin/java", "Client"], stdout=subprocess.PIPE)  
print p1.stdout.read()  
  
with open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-ship.txt", "r") as file:  
    shipRes=file.readline() #file.read()  
    shipConf=file.readline()  
shipConf = str(shipConf)  
shipRes=map(str,shipRes)  
shipRes2=shipRes[0]
```

Lines where port numbers and files are defined in each file:

newSerb.cgi - 71, 123, 162, 203

pay.cgi - 130, 183, 224, 267

b2b.cgi - 49 (this programs is for STAFF to be able to order to other companies)

6. Website Demo/Test

6.10 Registering - Transaction 0: Individual Account

Database tables before registering (query ordered by highest id's to show newest rows):

Customer Table before registering:

```
[MariaDB [b_f19_06_db]]> select * from customer_t order by customer_id desc limit 3;
+-----+-----+-----+-----+-----+-----+-----+
| customer_id | password | email           | customer_name | customer_type | customer_address | debit_card_num |
+-----+-----+-----+-----+-----+-----+-----+
| 120057    | habitat   | kevin@notmail.com | habitat       | Business      | 123 street      | 2147483647   |
| 120056    | v          | v                | v             | Business      | v               | 0              |
| 120055    | nn         | nn               | nn            | Individual    | nn              | 0              |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Registering:

Click On register on top
of navigation bar:



Fill out form entirely. For individual, click on the individual button on top. Website checks for already existing emails in database, empty lines, and for matching Strings when confirming password:

The Kitty Cabin

Clothes for the Ultimate Cat Lover

Home About Login Logout Register Shop Cart

Business Individual Staff

Registration

Full Name:
Pete

Email:
p@gmail.com

Password:
p

Confirm Password:
p

Address:
123 Easy Street

Submit

Click submit and click to login to start shopping:

The Kitty Cabin

Clothes for the Ultimate Cat Lover

Home About Login Logout Register Shop Cart

Congratulations! You Have Successfully Registered Pete!

Please login to start shopping:

[Login](#)

Let's look at our Customer Table now. Notice our new customer, Pete, at the very top:

```
MariaDB [b_f19_06_db]> select * from customer_t order by customer_id desc limit 3;
+-----+-----+-----+-----+-----+-----+-----+
| customer_id | password | email           | customer_name | customer_type | customer_address | debit_card_num |
+-----+-----+-----+-----+-----+-----+-----+
| 120058     | p         | p@gmail.com    | Pete          | Individual    | 123 Easy Street |      0        |
| 120057     | habitat   | kevin@notmail.com | habitat       | Business      | 123 street     | 2147483647   |
| 120056     | v         | v               | v             | Business      | v              |      0        |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (1.58 sec)
```

6.2 Logging in - Transaction 1

Now login. Use same email and password created during registration. Form connects to login.cgi program which checks customer_t in database to verify user exists:

The screenshot shows a web browser window with the URL `cs.newpaltz.edu/e/b-f19-06/`. The page has a light orange header with the text "The Kitty Cabin" and "Clothes for the Ultimate Cat Lover". Below the header is a navigation bar with links for Home, About, Login, Logout, Register, Shop, and Cart. The main content area is titled "Login" and contains two input fields: "Email:" with the value `p@gmail.com` and "Password:" with the value `p`. A "Submit" button is located below the password field.

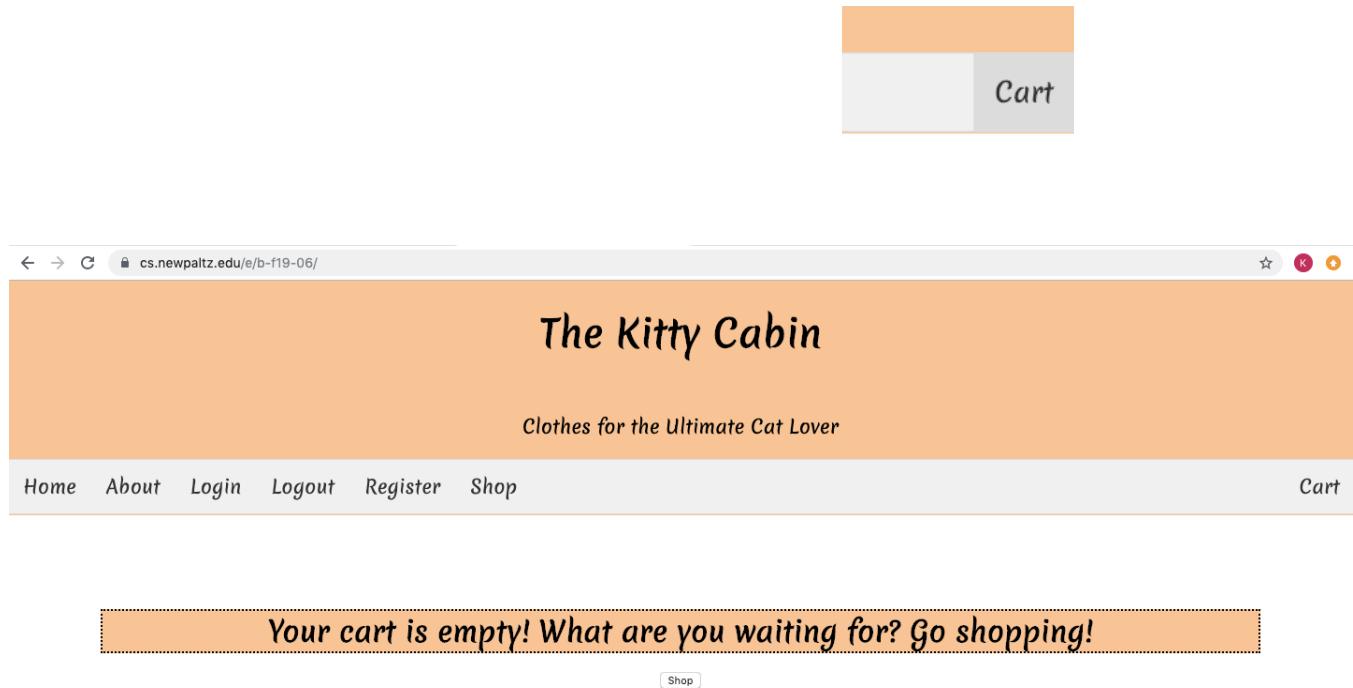
Click Submit and...

The screenshot shows the same web browser window after the login form has been submitted. The page now displays a welcome message "Welcome Back Pete!" in a light orange banner at the bottom of the main content area.

Success!

6.3 Viewing Items - Transaction 2

Let's check out our cart. It should be empty:

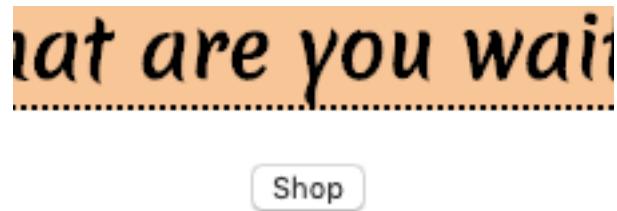


The screenshot shows a web browser window for 'The Kitty Cabin'. The title bar says 'cs.newpaltz.edu/e-b-f19-06/'. The main content area has a light orange background. At the top, it says 'The Kitty Cabin' and 'Clothes for the Ultimate Cat Lover'. Below that is a navigation bar with links: Home, About, Login, Logout, Register, Shop, and Cart. In the center, there is a large orange box containing the text 'Your cart is empty! What are you waiting for? Go shopping!' with a 'Shop' button below it.

Inside the cart table in the database. Right now Pete (id: (120058) doesn't have anything in his cart:

```
MariaDB [b_f19_06_db]> select * from cart;
+-----+-----+-----+-----+
| cart_id | customer_id | item_id | Quantity |
+-----+-----+-----+-----+
| 123185 | 120019 | 121001 | 1      |
| 123195 | 120022 | 121002 | 1      |
| 123219 | 120056 | 121000 | 1      |
| 123221 | 120048 | 121004 | 1      |
+-----+-----+-----+-----+
4 rows in set (3.62 sec)
```

Let's go shopping! You can click the Shop button on top or the Shop button in your empty cart page:



4 out of our 5 cool items:

A screenshot of a website for "The Kitty Cabin". The header includes "The Kitty Cabin", "Clothes for the Ultimate Cat Lover", and navigation links for Home, About, Login, Logout, Register, Shop, and Cart. A banner at the top says "Check out these cool items!". Four items are listed:

- Harry's Cool Shirt**: Price: 19.95, Item: Harry's Cool Shirt, View Item
- Harry's Cooler Shirt**: Price: 19.95, Item: Harry's Cooler Shirt, View Item
- Jerry's Sox**: Price: 3.95, Item: Jerry's Sox, View Item
- A pair of black pants with cat designs.

Click on view item to get a closer look and add to our cart. We'll click on "Harry's Cool Shirt":

The screenshot shows a web browser window with the URL <http://cs.newpaltz.edu/e/b-f19-06/>. The page title is "The Kitty Cabin" and the subtitle is "Clothes for the Ultimate Cat Lover". The navigation menu includes Home, About, Login, Logout, Register, Shop, and Cart. A banner at the top says "Harry's Cool Shirt". Below the banner is a white t-shirt with a black cat sitting on it. Product details below the image show Price: 19.95, Size: One Size Fits All, and Item ID: 121000. There is also an "Add to Cart" button.

6.4 Adding to Cart - Transaction 3

Change the quantity next to the "Add to Cart" button if would like to buy more than 1 item. We're going to buy 3 in this case and click "Add to Cart":

The screenshot shows a web browser window with the URL <http://cs.newpaltz.edu/e/b-f19-06/>. The page title is "The Kitty Cabin" and the subtitle is "Clothes for the Ultimate Cat Lover". The navigation menu includes Home, About, Login, Logout, Register, Shop, and Cart. A banner at the top says "Cart Was Updated!".

Viewing our cart again:

The screenshot shows a web browser window for 'The Kitty Cabin' website. The header features the store's name 'The Kitty Cabin' and a subtitle 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links for Home, About, Login, Logout, Register, Shop, and Cart. A dotted-line box highlights the 'Cart Page' section. Inside this box, there is an image of a white t-shirt with a black cat graphic. Below the image are details: Item: Harry's Cool Shirt, Price: 19.95, Size: One Size Fits All, and Quantity: 3. There is also a 'Remove Item' button.

6.5 Removing items from Cart - Transaction 3

Let's Remove 2 item. Either enter 2 or click on the arrows that pop up once you click on the quantity box. Click on the Remove Item box:

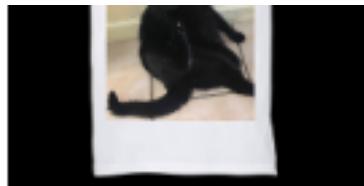
The screenshot shows the same web browser window for 'The Kitty Cabin' website. The 'Cart Page' section now displays a message 'Item Removed, Cart Updated' within a dotted-line box. The rest of the page content, including the header, navigation bar, and product listing, remains the same as in the previous screenshot.

Inside our cart table now that we added items for Pete (id 120058):

```
[MariaDB [b_f19_06_db]]> select * from cart;
+-----+-----+-----+-----+
| cart_id | customer_id | item_id | Quantity |
+-----+-----+-----+-----+
| 123185 | 120019 | 121001 | 1 |
| 123195 | 120022 | 121002 | 1 |
| 123219 | 120056 | 121000 | 1 |
| 123221 | 120048 | 121004 | 1 |
| 123222 | 120058 | 121000 | 1 |
+-----+-----+-----+-----+
5 rows in set (2.42 sec)
```

6.6. Checking Out - Transaction 4

To checkout, go back to cart and click on the Checkout button:



Item: Harry's Cool Shirt

Price: 19.95

Size: One Size Fits All

Quantity: 1

Change the Shipping Address and/or Bank Account Number if needed:

← → C ⓘ cs.newpaltz.edu/e/b-f19-06/ ⭐ K +

The Kitty Cabin

Clothes for the Ultimate Cat Lover

Home About Login Logout Register Shop Cart

Checkout

1 Shipping Information

Shipping Address:

Shipping Method: 1 Standard 2 Express

2 Payment Information

Bank Account Number:

3 Review Items

 Harry's Cool Shirt Price: 19.95 Size: One Size Fits All Quantity: 1

Click on the Review Order button to continue with the checkout process:

The screenshot shows a web browser displaying the 'The Kitty Cabin' website. The header features the store name 'The Kitty Cabin' and a subtitle 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links for Home, About, Login, Logout, Register, Shop, and Cart. A large orange call-to-action button labeled 'Review Order' is centered on the page. To its left, shipping information is listed: 'Shipping: Pete', '123 Easy Street', 'Ship Method: 1', and 'Bank Account: 1234567'. Below this, under 'Items:', there is a thumbnail image of a black t-shirt with a white cat graphic. Next to it, the item details are shown: 'Harry's Cool Shirt Price: \$19.95 Size: One Size Fits All Quantity: 1'. The total price is listed as 'Total: \$19.95'. At the bottom right of the page are two buttons: 'Place Order' and 'Back to Cart'.

6.7 Confirm Order - Transaction 5

Either click on the Place Order or Back to Cart button to go back to your cart. We'll click on the Place Order button to finish checking out:

The screenshot shows the same website layout as before, but the central content area now displays a confirmation message. The heading 'Confirmation Page' is centered above a large orange box containing the text 'SUCCESS! Thank You for your order!'. Below this box, a smaller note states 'Your Order Number is: 1200122602'.

```
Connected to wyvern.cs.newpaltz.edu at port: 11194 SENT File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-ship.txt to the Serverwyvern.cs.newpaltz.edu at port : 11194 Disconnected DONE.
Connected to wyvern.cs.newpaltz.edu at port: 11195 RECEIVING data from the Server at port : 11195 and write it to File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-ship.txt at port : 11195
Disconnected DONE. p 120058 888888 1234567 19.95 Connected to wyvern.cs.newpaltz.edu at port: 11198 SENT File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-bank.txt to the
Serverwyvern.cs.newpaltz.edu at port : 11198 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11199 RECEIVING data from the Server at port : 11199 and write it to File
/var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-bank.txt at port : 11199 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11196 SENT File /var/www/ebusiness/b-f19-
06/html/files/ClientSocket/Q-major.txt to the Serverwyvern.cs.newpaltz.edu at port : 11196 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11197 RECEIVING data from the Server at
port : 11197 and write it to File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-major.txt at port : 11197 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11198 SENT File
/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Q-it.txt to the Serverwyvern.cs.newpaltz.edu at port : 11198 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11199 RECEIVING
data from the Server at port : 11199 and write it to File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/A-it.txt at port : 11199 Disconnected DONE. True
```

SUCCESS! Thank You for your order!

Your Order Number is: 1200122602

If your order went through successfully, you will see the above page with a unique Order/Confirmation Number. The text above is showing how the ports are connecting to the various servers necessary (Bank, Shipping, Mayor and IT Services) to have a successful checkout. Let's check out our cart table now:

```
[MariaDB [b_f19_06_db]]> select * from cart;
+-----+-----+-----+-----+
| cart_id | customer_id | item_id | Quantity |
+-----+-----+-----+-----+
| 123185 | 120019 | 121001 | 1 |
| 123195 | 120022 | 121002 | 1 |
| 123219 | 120056 | 121000 | 1 |
| 123221 | 120048 | 121004 | 1 |
+-----+-----+-----+-----+
4 rows in set (0.21 sec)
```

Notice how Pete's id is no longer there. The order_t is now updated too. Notice Pete's order (customer_id 120058 - I had to redo screenshots to get the url on top hence the two orders from our friend Pete):

```
MariaDB [b_f19_06_db]]> select * from order_t where customer_id = 120058;
+-----+-----+-----+-----+-----+-----+-----+-----+
| order_id | customer_id | item_id_ordered | quantity | ship_method | ship_address | debit_card_num | order_conf |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 122594 | 120058 | 121000 | 1 | 1 | 123 Easy Street | 0 | 1200122594 |
| 122602 | 120058 | 121000 | 1 | 1 | 123 Easy Street | 0 | 1200122602 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

If you go back to your Cart on the website, it should be empty again:

The screenshot shows a web browser displaying the homepage of 'The Kitty Cabin'. The header features the site name 'The Kitty Cabin' and a tagline 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links for Home, About, Login, Logout, Register, Shop, and Cart. A large orange banner at the bottom of the page contains the text 'Your cart is empty! What are you waiting for? Go shopping!'.

The BankRequest_t, ShippingRequest_t, ITRequest_t and TaxRequest_t tables are also updated.
Examples showing Bank Requests and Shipping Requests:

The BankRequest_t:

```
[MariaDB [b_f19_06_db]]> select * from BankRequest_t order by CustomerId desc limit 2;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | PayeeAccount | PayorAccount | Amount |
+-----+-----+-----+-----+-----+-----+
| 120058 | p | 4191234567 | 888888 | 1234567 | 19.95 |
| 120055 | nn | 4191234567 | 888888 | 89 | 19.95 |
+-----+-----+-----+-----+-----+
2 rows in set (0.10 sec)
```

The

ShippingRequest_t:

```
[MariaDB [b_f19_06_db]]> select * from ShippingRequest_t order by CustomerId desc limit 2;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | ItemIDOrdered | ShipMethod | ShipAddress |
+-----+-----+-----+-----+-----+-----+
| 120058 | p | 2191234567 | 121000 | 1 | 123 Easy Street |
| 120055 | nn | 2191234567 | 121001 | 1 | nn |
+-----+-----+-----+-----+-----+
2 rows in set (0.14 sec)
```

7. B2B Demo/Test

7.1 Registering: Business Account

Registering:

Click On register on top
of navigation bar:



Click on Business Box and fill out form completely. Like Individual form, register.cgi program will check for empty lines, already existing emails in database and matching Strings when confirming password:

The screenshot shows a web browser window for 'The Kitty Cabin'. The title bar says 'cs.newpaltz.edu/e/b-f19-06/'. The main content area has a light orange background with the text 'The Kitty Cabin' and 'Clothes for the Ultimate Cat Lover'. Below this is a navigation bar with links: Home, About, Login, Logout, Register, Shop, Cart, and a dropdown menu with 'Business' (selected), 'Individual', and 'Staff'. The main content area is titled 'Registration'. It contains a form with fields for Full Name (Cool Biz), Business Email (coolBiz@gmail.com), Password (c), Confirm Password (c), Shipping Address (321 Cool Road), and Bank Account Number (654321). A 'Submit' button is at the bottom of the form.

Full Name:
Business Email:
Password:
Confirm Password:
Shipping Address:
Bank Account Number:
Submit

Click Submit button and success! Click Login:

The screenshot shows a web browser window for 'The Kitty Cabin'. The title bar says 'cs.newpaltz.edu/e/b-f19-06/'. The main content area has a light orange background with the text 'The Kitty Cabin' and 'Clothes for the Ultimate Cat Lover'. Below this is a navigation bar with links: Home, About, Login, Logout, Register, Shop, Cart, and a 'Login' link. A horizontal bar at the bottom of the page displays the message 'Congratulations! You Have Successfully Registered Cool Biz!' in bold black text.

Congratulations! You Have Successfully Registered Cool Biz!

Please login to start shopping:
Login

Notice the customer_t table in our database now with Cool Biz:

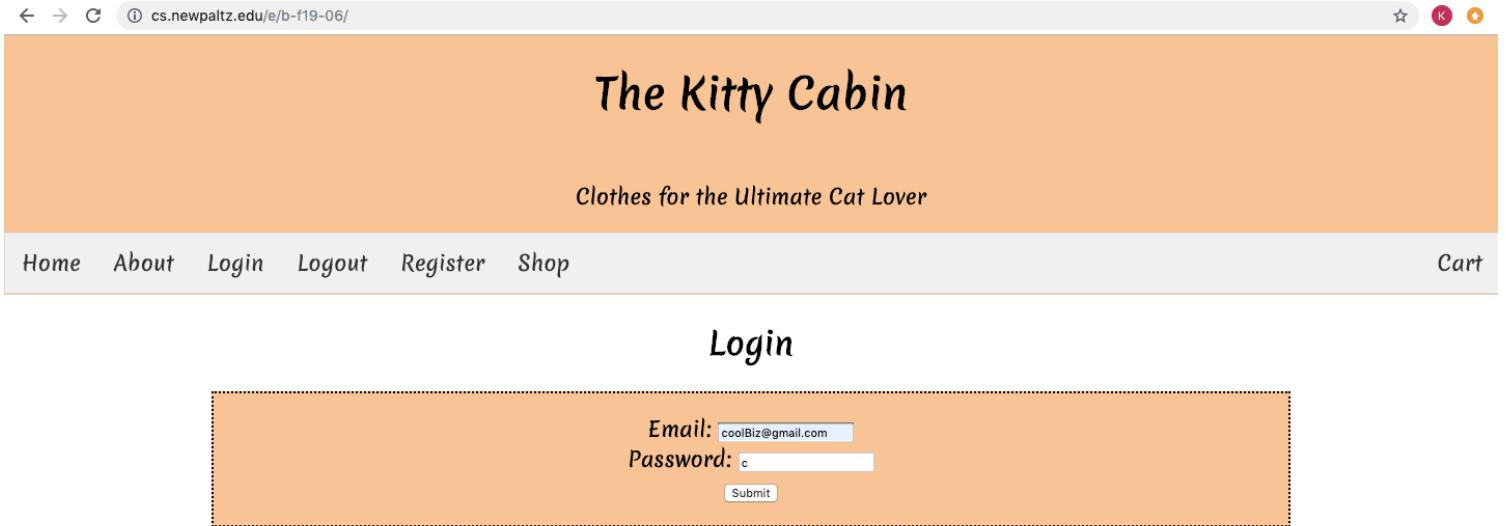
```
[MariaDB [b_f19_06_db]]> select * from customer_t order by customer_id desc limit 3;
```

customer_id	password	email	customer_name	customer_type	customer_address	debit_card_num
120059	c	coolBiz@gmail.com	Cool Biz	Business	321 Cool Road	654321
120058	p	p@gmail.com	Pete	Individual	123 Easy Street	0
120057	habitat	kevin@notmail.com	habitat	Business	123 street	2147483647

3 rows in set (0.00 sec)

7.2 Logging In

Enter email and password created when registering:

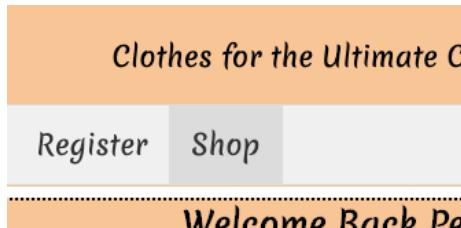


The screenshot shows a web browser displaying the login page for 'The Kitty Cabin'. The page has a light orange header with the title 'The Kitty Cabin' and a subtitle 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links for Home, About, Login, Logout, Register, and Shop. On the right side of the navigation bar is a 'Cart' icon. The main content area is titled 'Login' and contains a form with two input fields: 'Email:' and 'Password:', each with a corresponding text input box. A 'Submit' button is located below the password field. The entire form is enclosed in a dotted orange border.

Upon logging in successfully after verifying through customer_t table in database, directions on how to connect to The Kitty Cabin server will pop up. Follow them to purchase an item:

The screenshot shows a web browser window for 'The Kitty Cabin' website. The URL is 'cs.newpaltz.edu/e/b-f19-06/'. The page title is 'The Kitty Cabin' with the subtitle 'Clothes for the Ultimate Cat Lover'. The navigation bar includes links for Home, About, Login, Logout, Register, Shop, and Cart. A welcome message 'Welcome Back Cool Biz!' is displayed. The main content area contains instructions for B2B Ordering, including Customer ID Number (120059), Data Port (11120), Reply Port (11121), and file paths for Answer File and Question File. It also lists steps for ordering: 1. Your Customer ID Number provided above, 2. Your password you created when registering, 3. The Item ID number of the item you are purchasing (with a note about finding it in the catalog), 4. The quantity you are purchasing, and 5. Your desired shipping method (1 for standard, 2 for express). An example order is shown: 120000 kittyCabPass7! 121001 3 1. The page ends with a note about account types and a contact email address.

To Find the item id for an item to order, first, click on the Shop button on the navigation bar:



Click on View Item button to find item id. We'll check out Harry's Cooler Shirt:

The screenshot shows a web browser window with the URL cs.newpaltz.edu/e/b-f19-06/. The page title is "The Kitty Cabin" and the subtitle is "Clothes for the Ultimate Cat Lover". The navigation bar includes links for Home, About, Login, Logout, Register, Shop, and Cart. A banner at the top says "Check out these cool items!". Below the banner are four product cards:

- Harry's Cool Shirt**: Price: 19.95. Item: Harry's Cool Shirt. [View Item](#)
- Harry's Cooler Shirt**: Price: 19.95. Item: Harry's Cooler Shirt. [View Item](#)
- Jerry's Sox**: Price: 3.95. Item: Jerry's Sox. [View Item](#)
- Black Cat Leggings**: Price: 24.95. Item: Black Cat Leggings. [View Item](#)

Notice the Item ID: 121001 line. 121001 is the number needed to know what item you want to order through a B2B.

← → ⌂ ⓘ cs.newpaltz.edu/e/b-f19-06/

The Kitty Cabin

Clothes for the Ultimate Cat Lover

Home About Login Logout Register Shop

Harry's Cooler Shirt



Price: 19.95

Size: One Size Fits All

Item ID: 121001 Add to Cart

7.3 Order Through Staff B2B

To show how to order to The Kitty Cabin using a B2B, let's create a Staff account to have access to the B2B form:

The screenshot shows a web browser window for 'The Kitty Cabin' website. The URL is 'cs.newpaltz.edu/e/b-f19-06/'. The page has a light orange header with the title 'The Kitty Cabin' and the subtitle 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links: Home, About, Login, Logout, Register, Shop, and Cart. Under the navigation bar, there are three radio buttons: Business, Individual, and Staff, with 'Staff' being selected. The main content area is titled 'Registration' and contains a form with fields for Full Name (Kaitlin), Email (k2@gmail.com), Password (k), and Confirm Password (k). A 'Submit' button is at the bottom of the form.

Success once more!

The screenshot shows a web browser window for 'The Kitty Cabin' website. The URL is 'cs.newpaltz.edu/e/b-f19-06/'. The page has a light orange header with the title 'The Kitty Cabin' and the subtitle 'Clothes for the Ultimate Cat Lover'. Below the header is a navigation bar with links: Home, About, Login, Logout, Register, Shop, and Cart. A prominent message in a brown box at the bottom of the page reads 'Congratulations! You Have Successfully Registered Kaitlin!'. Below this message, there is a note: 'Please login to start shopping:' followed by a 'Login' button.

Look at our customer_t table again! Notice our new Staff member, Kaitlin:

```
[MariaDB [b_f19_06_db]]> select * from customer_t order by customer_id desc limit 3;
+-----+-----+-----+-----+-----+-----+-----+
| customer_id | password | email           | customer_name | customer_type | customer_address | debit_card_num |
+-----+-----+-----+-----+-----+-----+-----+
| 120060 | k       | k2@gmail.com    | Kaitlin      | Staff        | None            | 0              |
| 120059 | c       | coolBiz@gmail.com | Cool Biz     | Business     | 321 Cool Road   | 654321         |
| 120058 | p       | p@gmail.com     | Pete          | Individual   | 123 Easy Street | 0              |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.80 sec)
```

Click on login button. As with the other customer types, login information is verified through the customer_t database:

The screenshot shows a web browser displaying the homepage of 'The Kitty Cabin'. The header features the store's name in a large serif font and a subtitle 'Clothes for the Ultimate Cat Lover' in a smaller sans-serif font. Below the header is a navigation bar with links for Home, About, Login, Logout, Register, and Shop. On the far right of the navigation bar is a 'Cart' link. The main content area is titled 'Login' and contains two input fields: 'Email:' with the value 'k2@gmail.com' and 'Password:' with the value 'k'. A 'Submit' button is located below the password field.

The following is what appears once a Staff member logs in. Click on the B2B Form to order to another business through their Server:

The screenshot shows the same web browser after a staff member has logged in. The top navigation bar remains the same. A new message 'Welcome Back Kaitlin!' is displayed prominently in a yellow box. Below this, a link 'Ordering from another business? Click Below!' is shown above a small 'B2B Form' button.

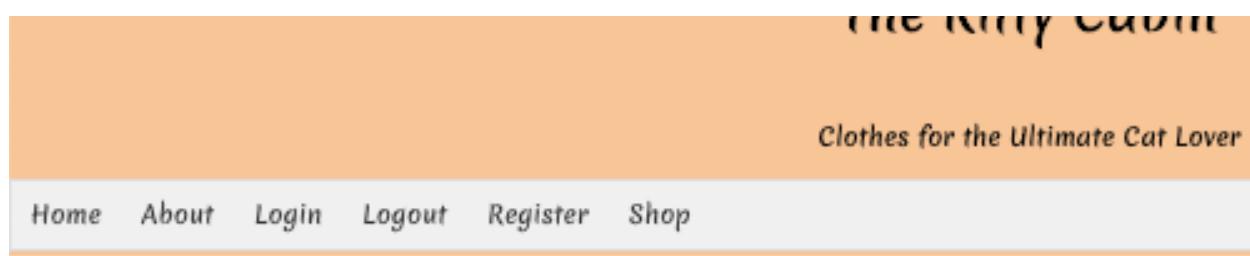
Input the necessary information following the business's B2B directions:

The screenshot shows a web browser window for "The Kitty Cabin". The URL is cs.newpaltz.edu/e/b-f19-06/. The page title is "The Kitty Cabin" and the subtitle is "Clothes for the Ultimate Cat Lover". The navigation menu includes Home, About, Login, Logout, Register, Shop, and Cart. The main content area is titled "Input necessary information for Ordering" and contains the following form fields:

- Customer ID: 120059
- Password: c
- Item/Service ID: 121001
- Quantity: 21
- Ports (separate by single space): 11120 11121
- Answer File want to read: /var/www/ebusiness/b-f

A "Submit" button is at the bottom of the form.

Once you click Submit, a Confirmation page should appear with a confirmation number. Unfortunately, something was off with the New Paltz Server at first, and I received a “Gateway Timeout”:



Gateway Timeout

The gateway did not receive a timely response from the upstream server or application.

After this issue occurred throughout the weekend, I decided to try to see if killing the background server process and then turning it back on would help. Luckily, it did and my confirmation went through:

The screenshot shows a web browser window with the URL cs.newpaltz.edu/e/b-f19-06/. The page title is "The Kitty Cabin" and the subtitle is "Clothes for the Ultimate Cat Lover". The navigation menu includes Home, About, Login, Logout, Register, Shop, and Cart. A message at the top of the content area reads: "Connected to wyvern.cs.newpaltz.edu at port: 11120 SENT File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Qc.dat to the Serverwyvern.cs.newpaltz.edu at port: 11120 Disconnected DONE. Connected to wyvern.cs.newpaltz.edu at port: 11121 RECEIVING data from the Server at port: 11121 and write it to File /var/www/ebusiness/b-f19-06/html/files/ClientSocket/Ac.dat at port: 11121 Disconnected DONE. True Transaction Successful!".

Let's check out our databases. I attempted to order twice which is why the tables have two orders from Cool Biz.

Order Table, order_t:

```
[MariaDB [b_f19_06_db]]> select * from order_t order by customer_id desc limit 3;
+-----+-----+-----+-----+-----+-----+-----+-----+
| order_id | customer_id | item_id_ordered | quantity | ship_method | ship_address | debit_card_num | order_conf |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 122595 | 120059 | 121001 | 2 | 1 | 321 Cool Road | 654321 | 1210122595 |
| 122596 | 120059 | 121001 | 2 | 1 | 321 Cool Road | 654321 | 1210122596 |
| 122594 | 120058 | 121000 | 1 | 1 | 123 Easy Street | 0 | 1200122594 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.28 sec)
```

Bank Request Table, BankRequest_t:

```
[MariaDB [b_f19_06_db]]> select * from BankRequest_t order by CustomerID desc limit 3;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | PayeeAccount | PayorAccount | Amount |
+-----+-----+-----+-----+-----+-----+
| 120059 | c | 4191234567 | 888888 | 654321 | 39.90 |
| 120059 | c | 4191234567 | 888888 | 654321 | 39.90 |
| 120058 | p | 4191234567 | 888888 | 1234567 | 19.95 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

Shipping Request Table, ShippingRequest_t:

```
[MariaDB [b_f19_06_db]]> select * from ShippingRequest_t order by CustomerId desc limit 3;
+-----+-----+-----+-----+-----+-----+
| CustomerID | Password | CRequestID | ItemIDOrdered | ShipMethod | ShipAddress |
+-----+-----+-----+-----+-----+-----+
| 120059 | c | 2191234567 | 121001 | 1 | 321 Cool Road |
| 120059 | c | 2191234567 | 121001 | 1 | 321 Cool Road |
| 120058 | p | 2191234567 | 121000 | 1 | 123 Easy Street |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

7.4 Inside StaffB2B CGI program

Ordering to another server works through the b2b.cgi program. Environment Variables are taken from the staffB2B form. Those are used to write into necPorts.txt to change the ports, flag files, and As and Qs files when connecting to another business's server (in this case, The Kitty Cabin). The answer from the company is then written to Ac.dat file in the ClientSocker folder under the files folder. Finally, the response is read. 0 is successful, anything else is unsuccessful transaction. The following is what is inside the b2b.cgi program (next page):

```

custId = myForm.getvalue('cusId')
custPass = myForm.getvalue('cusPass')
itemID = myForm.getvalue('itemID')
quan = myForm.getvalue('quant')
ports = myForm.getvalue('port')
asFile = myForm.getvalue('asfile')

db = MySQLdb.connect("localhost", "b_f19_06", "v5qteo", "b_f19_06_db")

myCursor = db.cursor()

##may not need since getting from webForm
print "Content-type:text/html\n\n"
print """
<html>
<body>
"""

if 'HTTP_COOKIE' in os.environ:
    cookie_string=os.environ.get('HTTP_COOKIE')
    c=Cookie.SimpleCookie()
    c.load(cookie_string)

    try:
        cData=c["custId"].value
    except KeyError:
        print "No cookie :("
print """
</body>
</html>
"""

b2bInfo = custId + " " + custPass + " " + itemID + " " + quan
b2bFile = bankFile = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Qc.dat", "w")
b2bFile.write(b2bInfo)
b2bFile.close()

#changing port:
changePort = open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/necPorts.txt", "w")
changePort.write(ports + " /var/www/ebusiness/b-f19-06/html/files/ClientSocket/f-Qc.dat /var/www/ebusiness/b-f19-06/html/files,
changePort.close()

p1 = subprocess.Popen(["/usr/bin/java", "Client"], stdout=subprocess.PIPE)
print p1.stdout.read()

with open("/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Ac.dat", "r") as file:
    b2bRes=file.read()
b2bRes=map(str,b2bRes)
b2bRes2=b2bRes[0]

transB2B = False
if(b2bRes2 == "0"):
    transB2B = True
else:
    print "Something wrong with B2B request"

print(transB2B)
if transB2B != True:
    print "Issue with B2B Transaction. Please Try Again."
else:

```

7.5 Inside Ac File

The Kitty Cabin sends over a 0 and confirmation number when transaction is successful:

```

[[hoffmank4@wyvern ClientSocket]$ cat Ac.dat
0
1210122596
[hoffmank4@wyvern ClientSocket]$ 
```

8. Developer Guide

8.1 How to Run/Setup The Kitty Cabin

In order for companies to have access to The Kitty Cabin, make sure **Server.java** is running as a background process with **java Server &**. The Server program can be found under **sockets** folder: **/var/www/ebusiness/b-f19-06/html/cgi-bin/sockets**

To check if The Kitty Cabin server is still running, enter **top** in Linux command line to see what is running. To end/kill the server background process, enter **top**, find the **pid**, and enter **kill pidNumber**.

Example:

If pid is 3047, enter **kill 3047** to send the process.

When sending an order to The Kitty Cabin, directions can be found after registering and logging in as a Business account. When connecting to The Kitty Cabin Server, business's Client program should change their port to send data to **11120**. To get a reply back from the server, change reply port to **11121**. Client program's answer file it is asking for should be **/var/www/ebusiness/b-f19-06/html/files/ClientSocket/As.dat**. The question file it is sending over to the server is **/var/www/ebusiness/b-f19-06/html/files/ClientSocket/Qs.dat**.

Please refer back to chapters 6 and 7 on how to use and access all features of The Kitty Cabin. All contents can also be found in **README.txt**.

8.2 Issues to Look Out For

If ports to the Bank, Shipping Company, Mayor, and/IT Services are closed, transactions can not be done successfully. In this case, error messages should indicate where the issue is coming from. Please contact these companies to resolve issues. Other companies may need to be used if issues with connections persist.

Sometimes connection to The Kitty Cabin server will be down. In this case, please contact the email under the About page for assistance in this matter.

8.3 Changes Needed When Moving Folders

CGI programs, **pay.cgi**, **newSerb.cgi**, and **b2b.cgi** have in them where to look for flag files and question and answer files. These are located under the **ClientSocket** folder under the **files** folder, **var/www/ebusiness/b-f19-06/html/files/ClientSocket**. If any of these files were to move, these changes must be reflected under **pay.cgi**, **newSerb.cgi** and **b2b.cgi**.

Many HTML files are connecting to other HTML files or cgi programs. Thus, if anything were to move, this must be reflected in all HTML files that are connecting them.

9. Challenges

9.1 Errors Encountered and How They Were Fixed

Moving everything to the shared folder

Since all CGI programs had to be in a different folder than the HTML files, all connections from an HTML file to a CGI program and vice versa had to be changed since they were now all in a different location

Connecting Server and Client files needed (As, Qs, flag files, etc.) in the Socket Programs

Again, files were moved to different locations and thus had to be changed in the java Client and Server programs. This was done by changing them to the absolute path to where those files are located. Unfortunately, I completely forgot to recompile my java Client and Server programs and spent way too much time trying to fix an issue that was technically already fixed. However, I eventually figured it out. Always recompile!

9.2 Other Challenges

Calling and Running a java program within a python/cgi program and vice versa

This was a little tricky but had some help from Stack Overflow which is referenced under references. In the end, it was rather simple and only about two lines of code.

Calling a java program within a python program:

```
p1 = subprocess.Popen(["/usr/bin/java", "Client"], stdout=subprocess.PIPE)
print p1.stdout.read()
```

Calling a python program within a java program:

```
try {
    Process p = Runtime.getRuntime().exec("python /var/www/ebusiness/b-f19-06/html/cgi-bin/newSerb.cgi");
}
catch (Exception e) {
    System.out.println(e);
}
```

10. Conclusions

I have learned so much more from this one project than I have learned from any other class. I had to implement everything I have learned so far, and more, and use it in this project. Before this, I had no idea how to use what I learned in my first semester in something that was useful and not just a bunch of tiny programs. I had to use HTML, css and JavaScript to create a user interface and connect all my pages together. I have created simple web pages, but it was never anything too detailed.

I never used CGI before, and even though I created a database in one of my past classes, I never connected it to anything. In this project, I learned how to connect my CGI programs to my databases and actually use that data and do something useful. Connecting everything together starting with my HTML pages, to my CGI programs, to my databases was the coolest thing in my mind.

I have also never used sockets before and did not even knew what they were before doing this project. Honestly, it sounded impossible, and I had no idea where to begin connecting my cgi programs with my Client programs. Calling a python program within my Server java program was something completely new as well. However, I figured it out with help from Professor Pham and some googling and made it work.

This was the hardest thing I have ever done, but it was the most fulfilling, and I am ecstatic to have something creative and useful to show to potential customers. I still plan on altering the look of my website even after it is graded and updating it as I learn new things.

11. References/Appendices

Using Cookies - used in all CGI programs to check who was using the website:

<http://raspberrywebserver.com/cgiscripting/using-python-to-set-retrieve-and-clear-cookies.html>

Java Server and Client Programs - <http://cs.newpaltz.edu/~phamh/basics/Sockets-Java/QAwithLocks/>

Running a python program from a Java program - used with Server.java program to connect to newSerb.cgi:

<https://stackoverflow.com/questions/10097491/call-and-receive-output-from-python-script-in-java>

Running a Java Program from a python program - used with pay.cgi, b2b.cgi and newSerb.cgi when calling my Client Server to connect to Bank, Shipping, Mayor and IT Services:

<https://docs.python.org/3/library/subprocess.html#module-subprocess>

<https://stackoverflow.com/questions/438594/how-to-call-java-objects-and-functions-from-cpython>

Clothes Pictures - used as photos for inventory_t table and to show item pictures on The Kitty Cabin website (cat photos are pictures of my own cats. I used Microsoft Paint to place them onto referenced pictures):

<http://www.clothesoutsale.com/juniors-racerback-tank-district-dt237.html> (blue tank top)

<http://cdn.zando.co.za/p/289404-4657-404982-2-detail.jpg> (white sock)

<http://clipart-library.com/images/8Txngyknc.jpg> (black shirt)

<http://www.obfuscata.com/wp-content/uploads/2017/10/t-shirt-template-20.jpg> (white shirt)

https://5fbb4b60e5a371522c26-c46478628be7be7c70f96ec65a31d1c7.ssl.cf3.rackcdn.com/Images/ProductImages/Black_Plain_Pyjama_Bottoms_56793_ab3.jpg (pajama bottom)