# Deliverable 3

**Angelica Longo**

**Project #1: MealStats**

**Fall 2020 – COP 4331C**

---

Contents of this Document



---

## Test Results

**Description of Test Environment**

The planned testing environment was supposed to utilize the Apache Tomcat Java servlet container tool inside Eclipse for its ability to render web pages. However, the front and back end of the final product was developed and designed using Java Swing GUI built from scratch in Eclipse, and the database management called "*phpMyAdmin* " was used for the implementation (import/export) of the MySQL database handled over a local web browser.

**Results of Individual Test Cases**

**I personally ran all test cases, no exact date on any test case, and testing was ran on Eclipse**

| Requirement ID | Requirement Description | Status | Notes |
|---|---|---|---|
| FUNCT-1 | Shall prompt user to login or create an account | Pass | |
| FUNCT-2 | Shall provide customers with a webpage of dish categories | Pass | Finished system is no longer a webpage |
| FUNCT-3 | Shall provide workers with a list of submitted orders from customers | Pass | |
| PE-1 | Shall run on all modern/smart devices which support internet access and contain a built-in web browser | Fail | Finished system is no longer a webpage. System now runs on any device that supports a JRE and the corresponding database is also needed |
| UHF-1 | Shall support users who are restaurant workers | Pass | |
| UHF-2 | Shall support users who are restaurant customers | Pass | |
| DOC-1 | Shall make use the of GitHub for visual control and configuration management | Fail | Design implementation was all done in Eclipse, and GitHub was only used for project submission |
| DATA-1 | Shall automatically add a 6.5% tax fee to every order submitted | Pass | |
| RES-1 | Shall operate on any web browser | Fail | Finished system is no longer a webpage (see requirement PE-1) |
| SECRTY-1 | Shall maintain a backup of all the customer's and worker's credentials | Pass | |
| QA-1 | Shall be personally documented which includes citations on reports, documentation in the code itself, testing, and applying improvements to finished features | Pass | |

**Conclusion**

The main requirements of the system operate as expected on the local testing machine. However, testing the system on another device has still not yet been perfected since the system currently mainly depends on a connection from local database that is setup in a web browser (this is due to no personal experience in working with databases prior to doing this project). Assuming that a database connection can be arranged on the device that wishes to operate the MealStats system (along with the support of a Java Runtime Environment), then the entire logic and operation of the system should be able to work perfectly fine.

# Build Instructions

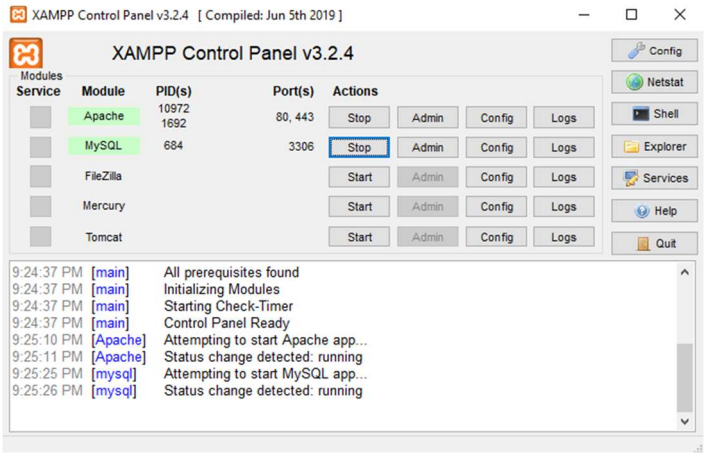
## Materials Required

All new users will need the following resources inorder for the MealStats system to run:

- XAMPP Control Panel
- Java (TM) Platform SE binary (a java runtime environment)
- MySQL connector file
- The MealStats database file
- MealStats executable JAR file
- An internet browser
- Link to the source code: https://github.com/angelica-zuly/MealStats

## Build Instructions

1. Install XAMPP Control Panel
   - This can be found at: https://www.apachefriends.org/index.html
   - Once launched, click "Start" for the **Apache** and **MySQL** modules and wait for them to turn green

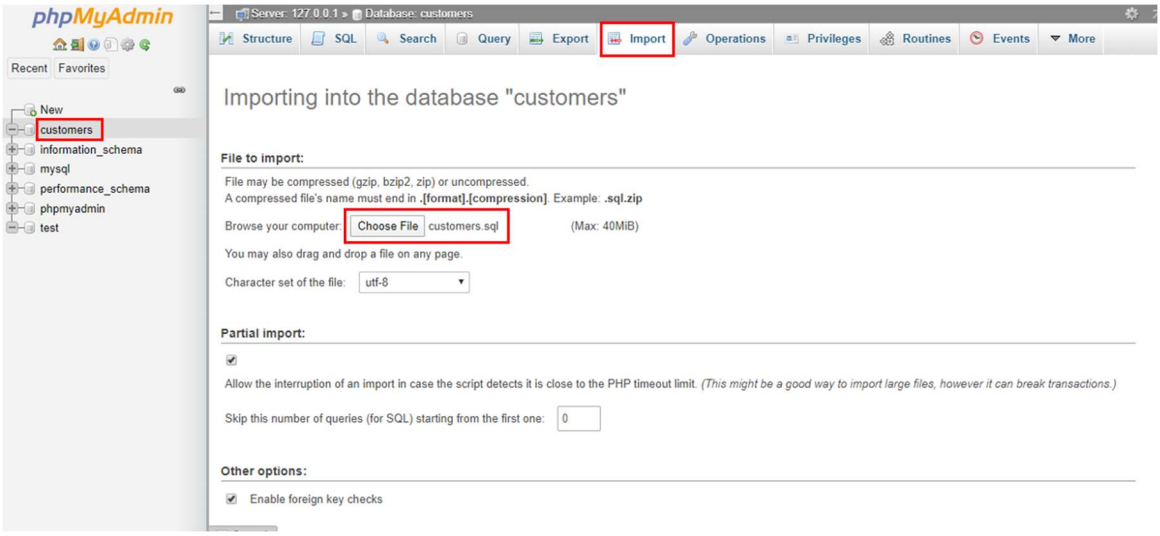



2. Download the **mysql-connector**
   - This can be found at: https://goo.gl/ftjWmK
   - Once downloaded, move the file (mysql-connector.jar) to the following directory:
   - Local Disk (C:) >> Program Files >> Java >> jre >> lib >> ext

3. Download the database (customers.sql) and MealStats JAR file (MealStats.jar)
   - These can be found at: https://github.com/angelica-zuly/MealStats

4. Open internet browser (Google Chrome is preferred) and type: "**localhost/phpmyadmin/**" into the URL
    - Create a new database by clicking "New" on the left side of the page, and type "***customers***" for the database name. Then click "Create" to finish.


- You will see the newly created *customers* database appear on the left. Click on it, and then click on the "Import" tab at the top banner.
- Click on the button that says, "Choose File" to import the database file mentioned above (customers.sql), then scroll down and click "Go" to finish.




5. You have now successfully setup the customers database. You will see that the "orders" and "users" tables have been imported under the customers database on the left, and you can now launch **MealStats** (the MealStats.jar mentioned above).

# Individual Lessons Learned

## Assessment of Project

1. ***In your opinion, what was the hardest part of this project? What was the easiest part of the project?***

   There are two main hard parts of this assignment. First one being how to start the execution of a web/mobile application with absolutely no experience given the time frame, and the second one being the implementation of a database connection from any device that uses the system. Given that the majority who are taking this class has some sort of programming experience, I would say that the easiest part of the assignment was coding the main logic and operation requirements of the project.

## Lessons Learned

2. ***What technical and non-technical lessons did you learn on this project (positive or negative) that you may be able to use in the future?***

   While working on this project I learned a lot about front and back-end development using Java Swing. I found myself spending a lot of time on small issues such as:

   - Selecting, deleting, and updating data from an SQL database (importing and exporting data took quite some time to learn and debug)
   - Disabling the default cell editor when removing a row from a table that contains and active item listener (this error would mess up my display and sadly took an entire day to figure out)
   - Creating a class to do all the “service utility functions”, which helped to organize getting and setting user data
   - Creating a unique order ID by using the system’s current time (I was having an issue with duplicate ID numbers each time the system restarts)
   - How to export a JAR file

## Course Suggestions

3. ***What suggestions do you have for future students who take this course?***

   I would suggest spending a lot of time researching how to connect to a database, as well as using an already built web application framework instead of making the entire GUI from scratch.