Brendan Boyle, Andrew Taylor, Mark Kilgore, Dylan Vetter, George Moya

CIS 331 Section 003

Professor Ezell

12/10/2019

Code update: 12/10

Earlier this week we finished a fully functional UI without a database connection. With the UI alone, the javafx GUI ran perfectly, executing every command from creating Guests, Bookings, Employees, Rooms, Room Services, to editing personal information, and Room information. Stress testing included clicking buttons without sufficient data filled out, attempting to create invalid passwords, Room numbers; all of which our UI passed.

Starting this week, we began the implementation of a database connection to the UI. Currently, our DB clears each ArrayList, uses the refresh(tableName) functions to pull information from the database, and then add it to each respective ArrayList we have. Upon closing the program, the insert(tableName) functions pull information from the ArrayLists into the DB. If a table runs into a unique constraint error, the sendDBCommand function automatically calls an update(tableName) function which will update that specific instance object. This is fine because if we make changes to the object, those changes will be updated, and if we don’t make any changes, the object is updated with all the same information, changing nothing.

We ran into problems with the insert/update and refresh functions for Room Service which (for the first Value Guest you create, book a Room, and add a Room Service charge) it will not show the Room Service charges; however, if you create a second Value Guest (book a Room, add a Room Service charge) then the Room Service charge list (which only shows for a Value Guest) will be updated with the correct Room Service.

* Basically the Value Guest Room Service charge list works for the first given Value Guest, not the next created Value Guest, and works for the rest of the created Value Guests

The problem lies in our insert/update or refresh functions for Room Service and, with a couple more days of balancing finals and working on this, we would absolutely find the problem and be able to deliver a fully functional program.