# Syed Muhammad Dawoud Sheraz Ali

# BESE-5-B

# 111417

**SE312: Software Construction**

# Lab 4: Persistent Restaurant Reservation System

**Date: March 15th , 2017**

**Time: Wednesday (10:00 – 13:00), Wednesday (14:00 – 17:00)**

# Instructor: Fahad Ahmed Satti

**GitHub:** <https://github.com/SyedDawoud/University/tree/master/Software%20Construction/Labs>

**Introduction**

This lab is about the Reservation System for a Restaurant. The purpose of this lab is to introduce the concepts of Software into the existing world. In this lab, we will implement the Restaurant Reservation System. This system allows the user to reserve the place in restaurant before going and waiting. Based on already reserved tables and any other special people, the system will allocate the time slot for the user. User can check its timeslot any time. This lab is an extension to the lab3, in which back-end database will be introduced and all the data will be stored at the back-end, to be used later

**Approach & Analysis**

Before explaining the approach, there were some ambiguities related to how the system would actually work. Given functional requirements were not clear. Therefore, some assumptions have been made:

* First, the restriction on the active reservation is based on the tables. Each Table has a class and has list of slot. Slots are assigned to table based on what the number of people are.
* Second, the time slot is increased to an hour. I.e. there is difference of 1 hour between 2 consecutive reservations. This has been considered because we have to give priority to impromptu customers, which usually come in large number as compared to reservations.
* Third, only one Special/XL table reservation per day. The time has been fixed to 10:30PM for that.

Some approach has been explained in the assumptions. Here is some more explanation in this regard. First, since there is so much data, different classes have been made to incorporate such massive data. There are separate classes for **Restaurant, Reservation System, and Person who is placing the Reservation.** There is main class that will handle all the working. For reservation, first some constraints are checked.

* Number of people should be in interval [0,12]
* If there are no tables free, trying to place reservation will throw exception
* Trying to place special reservation more than once in a day will also cause an exception to be thrown.

Also, no checks have been introduced to check that whether user is inputting correct value or not. However, empty value check has been implemented. Also, items in order aren’t checked with the menu either.

At the back-end, MySQL database has been added. It will keep track of all the changes and any user related information. For the reservation, some changes have been done. To see the reservation, you need to have Staff tag account. Also, if user tries to make reservation, if he/she have an account, the reservation will be made. But if user is new, an automatic record insertion will be done for that user.

**Running the Code**

The code has been written in Java. Source code and the unit tests are in separate file. To run the code, some IDE supporting Junit must be installed (NetBeans is preferred). Include the given source files and test files. Some package configurations might be required. Otherwise, use the main.java file to run the code, which will open the GUI for user to interact with. Also, JDBC needs to be installed to communicate with the database provided.

















