# Department of Computing

**SE312: Software Construction**

**Class: BESE – 5 AB**

# 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

# 

# Lab 4: Persistent Restaurant Reservation System

## Introduction

In this lab the students have to update their own implementation of Restaurant Reservation System from lab 3b to achieve persistent storage.

## Objectives

After performing this lab students will be able to understand:

* Separation of concerns
* Change Management
* Data Management API (JDBC)

## Tools/Software Requirement

* Solutions should be made using C++, Java, Python, or C#.
* You can take help from internet but remember **no plagiarism.**

## Description

The management at the client restaurant wants to use a persistent storage mechanism to ensure that the reservations made by the customers the not lost. In this regard, you are to design a Data Management part for the application you made during lab 3. The management would also like to add a simple authentication mechanism for the application, so that only pre-registered and authenticated users can make a reservation. The management would also like to view some kind of logs, which can show them what everyone has been up to.

Start by separating the Data Management part from the business logic, by using a model class and associated objects for each entity in your system. Your business logic, should only communicate with these entities, which in turn can communicate with the database.

Add a new user model class, where a user’s username and password are kept. This information is used to authenticate the user before they can make a new local reservation. Every guest can make an anonymous reservation. Similarly only staff members, if authenticated, should be able to view the reservation details for any period of time.

Finally, at all critical points, where Data is being manipulated (look for any insert, update or delete queries) log the event, the username, and the time. For logging, you can either use a table in your DB or put all logs in a file.

Each student must, individually build the complete application on their own. Students must upload their solutions on LMS to qualify for evaluation.

## Lab Task

* Add a persistent layer to the Restaurant Reservation System, you made in the previous lab.
* If you haven’t done the previous lab, now is a good time to start.
* Add authentication for all local reservations.
* Log critical events in your application.
* Ensure your implementation is correct by checking the requirements from the previous Unit Tests.
* Ensure the authentication system works with the help of a new unit test.

## Deliverables

* Each submission is individual with the following composition:
  + Source Code
  + Unit Tests
  + Documentation(Introduction, Approach, Design, How to Run and Analysis)
  + Link to the public repo on GitHub
* Convert your submission files into a zip folder and name it as given below, finally upload the zip folder to LMS.
  + Name – Registration No. – Section

## Grade Criteria

This lab will be graded on the following rubric, with minimum marks 0 and maximum marks of 24:

