

# Simple Cafe Manager Assignment

The purpose is to get information about interviewee's knowledge level of industry standard frameworks and coding practices. The grading of the assignment will take into consideration the functionality of the application, cleanness and architecture of the code, quality of the UX and other factors.

## OVERVIEW

Assignment is to implement web based JAVA Application on cafe management software.

## FUNCTIONAL SPECIFICATION OF PROJECT

Application should allow creation and management of orders. The application has 2 distinct interfaces and menu layouts for cafe managers and waiters.

The domain entities of the application are:

1. User (types: Manager, Waiter) - fields' definition is up to implementer
2. Table - fields' definition is up to implementer
3. Order - fields' definition is up to implementer
4. Product - fields' definition is up to implementer
5. ProductInOrder - fields' definition is up to implementer

Application has the following 2 roles:

1) Manager - Is allowed to:

- Create Users
- Create Tables
- Create Products
- Assign Tables to Waiters, each Table can be assigned only to 1 waiter

2) Waiter - Is allowed to:

- See Tables assigned to him
- Create order for table (if that table does not have Order which is still Open)
- Create ProductInOrder for Order (For example It will contain orderid, productId and amount (2xCola))
- Edit ProductInOrder fields (for example amount, status (active, cancelled))
- Edit Order fields (for example status (open, cancelled, closed))

## IMPLEMENTATION DETAILS

Project should be implemented as web application (or Android app or Desktop app if decided before the test)

- Any web applications frameworks the testee feels comfortable with can be used (JSP, JSF, Struts, Wicket, Spring MVC, etc ...)
- Use of ORM for database related activities is highly recommended (Hibernate, iBatis, etc ...)
- Use of IOC (Inversion Of Control) containers for application initial startup and DI (Dependency Injection) is highly recommended (Spring, Google Guice, etc ...)
- Use of TDD (Test Driven Development) practices is encouraged (JUnit, etc ...)
- Please use the best coding practices you know (formatting, comments, etc ...)

## CREDENTIALS

### TEST COMPUTER CREDENTIALS

1. username - User
2. password - Welcome555

### MYSQL DATABASE CREDENTIALS

1. Username: root
2. Password: root

### MS SQL DATABASE CREDENTIALS

1. Username: sa
2. Password: Welcome555

**SFL Wi-Fi: SFLaccess12#**