# Assessment Parameters

* Complete flow of the application  with validation and exception handling -70%
* Comments/best practice, coding standards – 10%
* Execution of the application (Output) – 20%
* ScreenShot should be submitted along with the solution
* The solution(Project) created by the trainee should have the name like AppName\_Empid Ex:ABCCorp\_675467
* Code with compilation errors will not be considered for evaluation

**Ticket Management System**

**Problem Statement**

XYZ Helpdesk needs an application for ticket management. This application is for raising tickets ( by the employee ) . Other functionalities are out of scope.

Consider one HashMap TicketCategory Which contain categoryId as key and categoryName as value ;

Consider another HashMap TicketLog which will store the Ticket Details (ticket No as key and Ticket Object as value )

**Note : Use Math.random method to generate the TicketNo**

1. **Display appropriate menu**

This functionality starts with displaying "Welcome to ITIMD Help Desk".

It should display following Menu.

1. Raise a Ticket

2. Exit from the system

1. **Employee : Raise a Ticket Marks [60]**

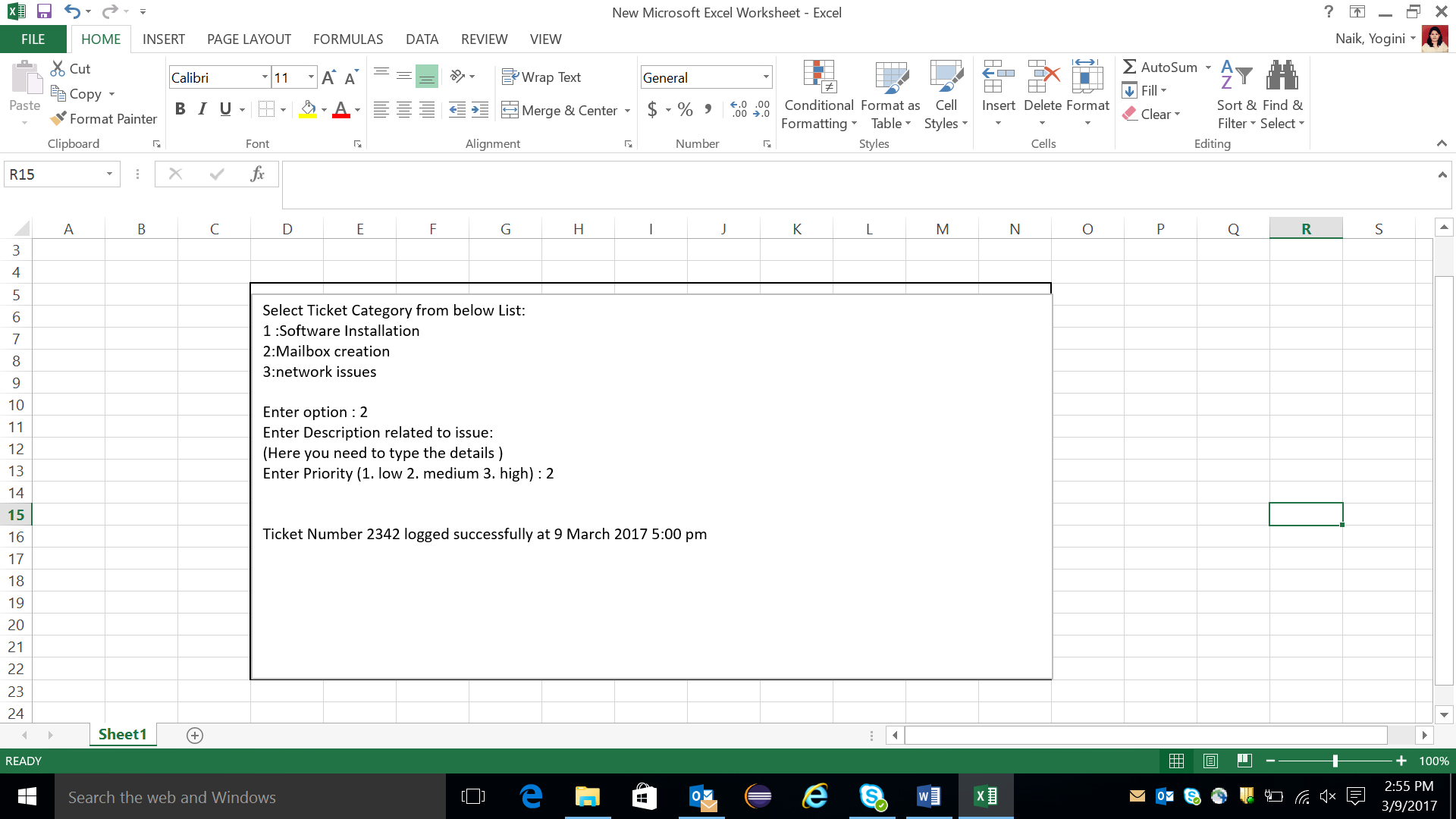
This functionality involves accepting details from the user and adding into TicketLog Map. Ticket category ( should be an existing one.).Ticket description ( must be entered ).Ticket priority ( must be either low/medium/high ). Ticket status should be set to “New”.

After adding the ticket details, display the following messages

Ticket number <generated\_ticket\_no> logged sucessfully at <date and time>.

**Date and Time should be current date and time.**

**Plese refer below screen shot(Category should come from Ticket\_Category table )**



* **Marks Distribution:**

|  |  |
| --- | --- |
| Selecting ticket category from ticketCategory and displaying it on console using DAO + service layer | 17 |
| Accepting the user details and generating the ticketNo using Math.random() method | 13 |
| Inserting the accepted user details in HashMap | 10 |
| Presentation | 10 |
| Service and Bean | 10 |
| **Total** | **60** |

1. **Exit**

When employee selects this option, user should be able to quit from application.  **[3 Marks]**

1. Write JUNIT test case for “raise a ticket functionality” of TicketDAOImpl class

**Marks[7]**

**Classes to be created**

**com.cg.tms.ui** // package containing main class

class MainUI{

public static void main(// to display the menu and accept the details from user

// create object for service class and execute the respective methods

}

**com.cg.tms.dto** //package containing all beans

public class TicketCategory{

private String ticketCategoryId;

private String categoryName;

// getter, setter,toString & constructor

}

public class TicketBean{

private String ticketNo;

private String ticketCategoryId;

private String ticketDescription;

private String ticketPriority;

private String ticketStatus;

private String itimdComments;

// getter, setter,toString & constructor

}

**com.cg.tms.service** // package with service layer class

public interface TicketService {

boolean raiseNewTicket(TicketBean ticketBean);

List<TicketCategory>listTicketCategory();

public class TicketServiceImpl implements TicketService{ …….. }

**com.cg.tms.dao -**// package containing data access class to perform database operations.

public interface TicketDAO {

boolean raiseNewTicket(TicketBean ticketBean);

List<TicketCategory>listTicketCategory();

}

public class TicketDAOImpl implements TicketDAO{

……………

}

**com.cg.tms.util -**// package containing data access class to perform utilityoperations

public class Util{

private static Map<String, String> ticketCategory=new HashMap<String, String>();

public static Map<String,String> getTicketCategoryEntries(){

ticketCategory.put("tc001","software installation");

ticketCategory.put("tc002","mailbox creation");

ticketCategory.put("tc003","mailbox issues");

return ticketCategory;

}

}

Add appropriate user defined exception classes required.