University of Technology, Jamaica
School of Computing and Information Technology
CIT3009 – Advanced Programming

Facilitator(s):	**Christopher Panther**		
Assessment:	**Group Assignment (No more than 2 persons per group)**		
Given week of:	**September 30, 2020**	**Due week of:**	**November 23**

The University of Technology (UTECH), Jamaica, is the premiere University in Jamaica. One of the outstanding features of UTECH is the efficiency with which students' Services are dealt with. However, the COVID-19 pandemic has drastically impacted the harmonious relationship the university's administrative staff has enjoyed with its students. There are now gaps between students' Services queries and their resolution. The university must now contend with numerous complaints from students. To maintain the efficiency in treating with student Services issues for which the university is known, the Students Services manager wants a query logging system that will be able to capture all student complaints and queries. Upon receipt of a complaint or query, a Student Services Agent will log the complaint or query and schedule a Student Services Representative to contact the student to rectify the issue remotely.

You are to build a TCP/IP socket-based Client / Server architecture application that will facilitate the above scenario. Your Client application must send requests to the server application as only the server application can establish connections to the database. The client application must consist of a GUI. The server application need not have a GUI. However, it may include graphical components to enhance user experience.

Your system must capture all data about Students, Student Services Representatives, Student Services Agents, Complaints and Queries.

System Functionalities

Authentication:

- Students should be able to log on to the system using their Student ID number and password to gain access to the dashboard

- Student Services Agents and Student Services Representatives (Employees) should be able to log on to the system using their Staff ID and password to gain access to the dashboard

Student Services

- Students should be able to choose the service they would like assistance with, on the dashboard; they should also be able to register the nature and details of their query or complaint, so that a Student Services Agent can access it at a later time.
- Students should be able to view all past complaints or queries in a list. For each complaint or query in the list, the last response date and who provided the response, should be shown.
- Students should also be able to view a specific complaint or query and all its associated responses.

Employee Services

- Employees should be able to view a list of services on the dashboard along with the number of resolved and outstanding queries and complaints.
- Employees should be able to view all student enquiries relating to a particular service.
- Student Services Representatives should be able to view a particular student enquiry and the details relating to that issue, so that they may prepare a response.
- When viewing specific students' queries or complaints, a Student Services Representative should see students' details to include:
  - Student ID,
  - Name (first name and last name),
  - Email address,
  - Contact number,
  - Type of issue, and
  - Details of the issue.
- Student Services Representatives should be able to respond to students' query and complaints.

Live Chat

- Student Services Reps should be able to indicate their availability for a live chat session, so that students can initiate a live chat session
- Student Services Reps should be able to participate in a live chat session with students, to address their issues.
- Students should be able to initiate a live chat with a Student Services Rep, to address their issue.

Development Tools

- Eclipse IDE (JDK 1.8 or higher)
- MySQL 8 current version (Administered with MySQL Workbench or PhpMyAdmin)
- MySQL Connector/J (JDBC)
- Hibernate 5 (Object Relational Mapping)
- Apache Log4J2 (Logging and Audit Trails)
- Git (Version Control)

Deliverables

- Group Report
  - Details of each group members contribution
  - Declaration of Authorship - one for each group member
- Application User Manuals
  - Student Services Employees User Guide (Document or Screen Recording)
  - Student User Guide (Document or Screen Recording)
- Server-side application
- Client-side application – Student
- Client-side application – Student Services Reps

Assessment

- Written Report, to include OOAD and Learning Trail for each group member, i.e., challenges faced, steps taken to overcome the challenges, and lessons learned.

- Application User Manuals
- 30 minutes grading interview – all group members must be able to fully present the project.

Milestone 1 – 10 Marks – Weight 20%

Database Design and Domain – 3 Marks

- Create Tables (Fields, Data Types, Constraints, Relationships)
- Normalized Tables (At least 3rd Normal form)
- Create Domain Objects (Including Inheritance)

Traditional Database Connectivity – 3.5 Marks

- CRUD operations (select with condition, multiple select, insert, update, delete)
- Managing Result Sets
- Manage and Log All Exceptions

Object-Relational Mapping – 3.5 Marks

- Setup ORM
- CRUD operations (select with condition, multiple select, insert, update, delete)
- Managing Result Sets
- Manage and Log All Exceptions

Milestone 2 – 10 Marks – Weight 20%

Event Logging – 5 Marks

- Setup Loggers
- Setup Appenders (Console and RollingFile)
- Setup Layouts
- Logging All Events (Information, Warnings)
- Logging All Exceptions (Errors and Fatal)

Version Control – 5 Marks

- Create Project Repository (Git Desktop Client, Eclipse Git or GitHub)

- Collaboration by group members using Git

Milestone 3 – 10 Marks – Weight 20%

TCP/IP Socket-based Networking and Object Serialization– 7 Marks

- Setup TCP/IP Client
- Setup TCP/IP Server
- Mechanisms to send serialized objects over network streams
- Log All Events
- Manage and Log All Exceptions

Threading – 3 Marks

- Thread TCP/IP Server Engine to manage multiple client requests
- Log All Events
- Manage and Log All Exceptions

Milestone 4 – 10 Marks – Weight 20%

Advanced Graphical User Interface – 6 Marks

- Create Parent Window
- Create Internal Forms for each database table (insert, select, update, delete)
- JTables with Table Models must be used
- Implement Menu Bars, Keyboard Short Cuts, Tool Tips, Mnemonics

Remote Method Invocation – 4 Marks

- Create Remote Interface
- Implement Remote Object

System Functionalities – 10 Marks: Weight 20%

- Users - Students & Reps, should be able to log into the system with their student ID and password, and Staff ID and password respectively **[5 marks]**

- Students should be able to choose the category of issue they need assistance with, and state the nature of the enquiry, so that a Rep may view the enquiry at a later time. **[5 marks]**
- Students should be able to:
    - View a past enquiry and all its associated responses. **[5 marks]**
    - View a list of past enquiries, showing in the list, the last date of response and the name of the person who provided the response. **[5 marks]**
- An employee / rep should be able to view a list of services from a dashboard, along with the number of resolved and outstanding enquiries. **[5 marks]**
- An employee should be able to view all students' enquiries by category. **[5 marks]**
- Employees should be able to view the enquiry details, and account information of a specific student, so that an appropriate response may be prepared. **[5 marks]**
- An employee should be able to respond to a student's enquiry. **[5 marks]**
- The system should contain a live chat feature, which will:
    - Allow students to initiate a live chat with employees. **[2.5 marks]**
    - Allow employees to engage students in a live chat session, to provide feedback on enquiry status. **[2.5 marks]**

Bonus

- The system should also contain a live video chat feature, which enables live video chat between students and employees. **[10 marks]**