20 October 2017

Nanga Systems GmbH \_ Friedrichstraße 15 \_ 70174 Stuttgart

Exercise 1: Java Development

Write a JAVA application that fulfils the following requirements. You will have to present your application as well as the source code written for it.

|  |  |
| --- | --- |
| Requirement no. | Req\_001 |
| The application shall be written and build in JAVA. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_002 |
| The application shall have a graphical user interface implemented with JavaFX. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_003 |
| The application shall be executable on Microsoft Windows. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_004 |
| The application shall have a function that runs “***tasklist***” as a console command. The output of this command shall be returned to the application and show the following information (of each task, sorted by used memory) within the graphical user interface:   * Name * PID * Used Memory | |

|  |  |
| --- | --- |
| Requirement no. | Req\_005 |
| The application shall provide a manner of handling different OS language settings which might affect the result of the “tasklist” command. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_006 |
| The application shall have a function that removes any duplicates from the listed tasks (assuming the name is the identifier – ignoring the PID). Tasks of the same name shall be grouped together and the used memory aggregated. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_007 |
| The application shall have a function that exports the cleaned list of tasks (see Req\_005) into a XML file abiding by the following scheme:  <tasks>  <task>  <name></name>  <memory></memory>  </task>  </tasks> | |

|  |  |
| --- | --- |
| Requirement no. | Req\_008 |
| The application shall utilize a SaveFileDialog form (or something equivalent) to give the user a method to choose the filename and location of the new xml file. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_009 |
| The application shall have a function to re-import the saved XML file and compare the contents to the data shown in the graphical user interface. Any changes shall be shown in the GUI. | |

|  |  |
| --- | --- |
| Requirement no. | Req\_010 |
| The application shall provide a function that exports the gathered data into Microsoft Excel and generate a chart about the memory usage of tasks. | |

Exercise 2: SQL statements

Premise:

A SQL database consists of the following tables:

Internal Users:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Employee ID | Username | Department | Job title | Location | Phone number | Mail address |

External Users:

|  |  |  |  |
| --- | --- | --- | --- |
| Employee ID | Username | Location | Phone number |

Locations:

|  |  |  |
| --- | --- | --- |
| Location ID | Location | Facility Manager |

Write a single SELECT statement that returns all users (internal and external) with their Employee ID, Username, Department, Location and Job title (where applicable) that work in a location managed by “Location Manager Alpha”.

Answer to SQL task:

SELECT iu.employee\_id, iu.username, iu.location,

eu.employee\_id, eu.username, eu.location

from internal\_users iu

JOIN locations l ON iu.location = l.location

JOIN external\_users eu ON eu.location = l.location

where iu.location = (SELECT location FROM locations WHERE facility\_manager = 'Location Manager Alpha');