| **LAB211 Assignment** |  |  |
| --- | --- | --- |
| **Lab 14** |  |
| **LOC:** | **180** |
|  |  |

**Title**

Task management program of CCRM project

**Background**

(Module extracted from TienPhong Bank, ebank project)

**Program Specifications**

Write a program to manage the task and task type for employees include function to delete:

* The type of task: (ID, Name contains the following data fixed:
  + ID Name

1 Code  
2 Test  
3 Design  
4 Review

* Task: ID, TaskTypeID, Requirement Name, Date(Aug-10-2021), Plan From, Plan To, Assignee, Reviewer
  + (ID = ID last task +1)
  + Plan From, Plan To calculate from 8h -> 17h30 ⬄ 8.0, 8.5, 9.0, 9.5 … -> 17.5.

***Function details:***

**Function 1:** Display a menu and ask users to select an option.

* Users run the program. The program prompts users to select an option.
* Users select an option, perform **Function** **2**.

**Function 2:** Perform function based on the selected option.

* Option 1: Add Task
  + Prompt user to input the information requested Task (TaskTypeID, Requirement Name, Date, From, Plan To Plan, Assignee, Expert)
  + Check for valid data with the conditions:
    - Check the TaskTypeID must exist (1-4).
    - Information must be valid date in the format M-d-yyyy.
    - Plan From must be less than Plan To and within 8 h-17 h 30 > 8.0, 8.5, 9.0, ⇔ 9.5 ...-> 17.5 .
  + Add a Task on the program .
  + Go back to the main screen.
* Option 2: Update Task
  + Check for valid Task to update
  + User can edit of the remaining information,  if information is “nope” then not change old information
* Option 3: Delete Task
  + Request input the ID of the task to delete.
  + Check for valid data with the conditions below:
    - Id must exist in the DB.
  + Delete the task.
  + To return to the main screen.
* Option 4: Show task
  + Show the task of descending according to the ID and the required format interface.
  + To return to the main screen.
* Option 5: Exit the program.

***Expectation of User interface:***

******

**Guidelines**

**Student must implement methods**

addTask

updateTask

deleteTask

getDataTasks

**in startup code.**

Uses try-catch to catch NullPointerException, NumberFormatException

Use SimpleDateFormat to handle date.

Use wapper classes to test the value number.

**Option 1:** Add the task.

* Implement function: public int add Task (String requirementName, String assignee, String, String, String taskTypeID expert date, String, the planTo planFrom double) throws Exception
  + input :
* requirementName: Name of the requirement
* assignee: task assigned to.
* reviewer: Review task.
* taskTypeID: task type.
* date: task performed date
* planFrom: Start time.
* planTo: End time.
  + Return value:
* id task
* Exception list

**Option 2:** Update the task.

* Implement function: public void update Task (String ID, String requirementName, String assignee, String, String, String taskTypeID expert date, String, the planTo planFrom double) throws Exception
  + input :
* ID: request enter the code. If it does not exist Code, the notification "code does not exist" and then repeat until it exists.
* requirementName: Name of the requirement
* assignee: task assigned to.
* reviewer: Review task.
* taskTypeID: task type.
* date: task performed date
* planFrom: Start time.
* planTo: End time.
  + Return value:
* Exception list

**Option 3:** Delete task.

* Implement function: public void deleteTask (String id) throws Exception
  + input :
* id: id task
* Return value: Exception list

**Option 4:** Show task.

* Implement function: public function settings getDataTasks ()
  + Return value: list of task