# Description:​

Create a simulation of a banking system that allows the user to create K

The client can have multiple accounts with different types and the clients should be able to:

1. Display their current balance
2. Deposit money
3. Draw money and (charge a fee)
4. Transfer money to other accounts within the bank
5. Pay utility bills
6. Other transactions you would like to add.

But, once the clients’ accounts created must be saved in a text file, so we don't need to enter their data anymore. But the system must be able to change their data if needed.

Always when the user wants to use the system must get a menu to choose the required option, and maybe after choosing the menu there is a sub-menu popup.

The system can do the following:

1. Let the user create accounts in different types
2. Let the user enter details about the clients and accounts according to the account type
3. Save all the details in a file.
4. Let the user be able to read and change the details
5. Let the user be able to perform the above transactions
6. Any other tasks you would like to add Note:

* There are different types of bank accounts, you should google it to know more
* You can’t use array you have to use ArrayList, so the number of clients or accounts is flexible.
* If you find you need further details or you have any questions about the system which are not mentioned here in the description, then you should answer by your selves.
* The description here is giving you the minimum requirements, only.

# How to submit

* Check on Moodle
* You will work within your groups.
* You must create a system report.pdf as a word then save as a PDF file that has:

○ Cover page showing your names and id’s

○ Index (table of contact)

○ Introduction a brief about the system

○ Full description of the system including the minimum requirements you have up at the above description **FLOW with ss**

○ Explain the classes with their attributes and methods (not code)

○ Explain exactly what the system can do (all tasks) Bullets

○ Explain how the system is working like you are creating a user manual  **Where user should be focused**

***Add description what method or statement does in eclipse and then add source code in the file.***

**○ The source code of all the system**

○ Some screenshots showing the errors or exceptions you get while you are testing the system, and explain how you could solve them.

**○ Some screenshots showing the system is working without errors**

**○ Name(s) of the file(s) the system is creating, and copy of their details**

○ Conclusion 5min video from each member: explain how you created the system, what did you do to collect more information and details about banking systems and different types of accounts. Explain the challenges you faced and how you could defeat them. Explain how you distribute the work among you, so explain what exactly each group member has done.

○ ***List of references***

# The rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Exceeds Expectations**  **(9-10 marks)** | **Meets Expectations**  **(6-8 marks)** | **Below expectations**  **(3-5)** | **Not meet expectations**  **(0-2)** |
| **The system description and tasks (requirements)** | The system is completely explained with a full description of the requirements and all tasks without any missing details and using a professional proper language and phrases | The system is completely explained with full description of the requirements and tasks without missing details | The system is completely explained with full description of the requirements and tasks but maybe some details are missed | The system is not completely explained and without a full description of the requirements and tasks |
| **The proposed system** | All the minimum requirements of the system that I explained in the project description are met completely, and some new features, tasks make sense and added value to the system | All the minimum requirements of the system that I explained in the project description are met completely, and some new features, tasks make sense and added value to the system | All requirements are done but not correctly nor completely | Some requirements are missing |
| **The system design** | The system is well designed in a way that shows the users and guides them for smooth and easy use and minimizes or avoids the data entry mistakes | The system is well designed in a way that minimizes or avoids the data entry mistakes | The system is well designed in a good way | The system design is not good enough |
| **The user manual** | Explains exactly how to use the system, with full details, using screenshots, or any other ways that make it easy for any user to use the system, and using proper language | Explains exactly how to use the system, with full details, that make it easy for any user to use the system, and using proper language | Explains how to use the system, with enough details, that make it easy for any user to use the system | Explains how to use the  system without enough details |
| **The source code** | The code is professionally clear, complete with no bugs, created by the group members, not downloaded from the web, well structured, with professional comments and exception handling | The code is clear, complete with no bugs created by the group members, not downloaded from the web, well structured, with comments and exception handling. | The code is clear complete with some bugs created by the group members, not downloaded from the web, with exception handling | The code is not clear not complete with some bugs or contains big parts downloaded from the web |
| **The classes** | All required classes are created and fully explained with their attributes and methods, using inheritance properly | All required classes are created with their attributes  and methods, using inheritance properly | All required classes are created d with their attributes and methods, | Not all required classes are created |
| **The report** | Well organized in a professional way with proper language and phrases, with all required parts, page numbers, and using of titles, subtitles perfectly. | Organized in a good way with proper language and phrases, with all required parts, pages numbers | Not well organized in a without proper language and phrases, with all required parts. | Not well organized Some part missing |
| **The Testing** | Showing major problems, failures, bugs, errors of the code, with full explaining how finding the solution | Showing major problems, failures, bugs, errors of the  code, and explain explaining them | Showing major problems,  failures, bugs, errors of the code | Not showing major problems, failures, bugs, errors of the code |
| **The system Demo** | The System is working with no bugs, and the group members all are able to answer the | The System is working with no bugs, and the group members all are able to | The System is working with no bugs | The System is not working with or has bugs |
|  |  |  |  |  |