CMPS 312 Mobile Application Development- Fall 2019

**Homework 3**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Student Id** |  |
| **Email** |  |

**Grading Rubric - - In the *Functionality* column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Points | Functionality\* | Quality of the implementation | Score |
| \*\*\*These criteria are just a guideline for grading.  However, you should meet all the requirements in the posted pdf document \*\*\* | | | | |
| Implemented the Login and Registration using shared preferences | 1 |  |  |  |
| Add  Able to add todo to the database  Update  Able to update a todo to in the database  Delete  Able to delete a single todo from the database  Delete All Todo  Able to delete all the todos in the database  Get All Todo  Able to get all the todos in the database | 3 |  |  |  |
| Database is linked with the UI of the application. This means all of the homework functionalities work properly. | 0.5 |  |  |  |
| Code properly organized [DAO, DBHelper etc…] | 0.5 |  |  |  |
| Total | 5 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | -5 |  |  |  |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***Not*** ***working*** (lose 40% of assigned grade) and ***Not done*** get 0. The remaining grade is assigned to the quality of the implementation. In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation. Quality includes the meaningful naming of identifiers, no redundant code, simple and efficient design, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation. **Marks will be reduced** forcode duplication, poor/inefficient coding practices, a poor naming of identifiers and unnecessary complex/poor user interface design.

## Testing evidence