**Milestone 2 Rubric – total 95 points**

**Instructor**: Provide 2-3 sentences of feedback to the student to explain the grade.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Unsatisfactory  0-69% | Satisfactory  70-85% | Excellent  86-100% | Total |
| Design Report (max 10 points):   * Cover Page and Planning * Technical Approach * ER Diagram * Wireframe Diagrams * Risks |  |  | X | 8.5 |
| Functionality (max 75 points):   * Login and Registration Modules * Profile Module * Admin Module (list, suspend, delete) * App Title/Menu/Navigation/Layouts * Screencast Demo |  |  | X | 72 |
| Efficiency and Misc. (max 10 points):   * N-Layer Architecture * OO Design Principles * Code in Source Control * Code with Comments * Code that is Clean (no junk, readable, formatted, etc.) |  |  | X | 10 |
| Total Points: |  |  |  | 90.5 |
| Comments:  Great job with your division of work.  ER Diagram: Relationships and PF/PK not shown or designed. You might want to break your education, and employment history down into more granular fields. What if you need to format / parse this information or 1000 chars is not enough? I know this makes your forms, db, etc. more complicated but give this some thought. You are missing somewhere to enter your contact info.  Wireframe: Fields on screen design do not match or have fields in the database.  Class Diagrams: No data types defined.  Functionality: Missing ability to enter contact info. You should prompt the admin for the delete user operation since this is destructive and someone could accidently click on this link.  Code: Overall very nice work. Are your queries case sensitive queries. The following is more of an FYI, that I am putting in as common feedback to all: Do not put session code in your DAO or Business Service. This is a presentation layer technology. Picture this. You reuse your DAO and Business Service with an iPhone front end and REST API’s. The HTTP Session cannot be created in this context because there are no Controllers and no MVC framework to manage the session cookie. Controller should interact with a Business Service class and not a Data Service (DAO). All in all this was very well designed and layered code. Overall very nice work getting blade incorporated into your Views with some nice exception handling in your layers. Keep commenting your code. | | | | |