### **Analysis**

The analysis is comprehensive, covering problem description, user requirements, and research on existing solutions. It clearly identifies the gap the project intends to fill and documents user needs through a questionnaire, indicating a fully scoped analysis.

The objectives for the proposed system are specific and cover a wide range of functionalities that the application aims to provide​​. However, for the objectives to be considered fully measurable, they should ideally include more quantifiable criteria or benchmarks to assess whether each objective has been met successfully. For example:

* **Login Page**: The objective is specific in describing the functionality of the login and registration pages, but it could be made measurable by stating the expected time for a user to complete the registration or login process.
* **User Experience Level Determination**: This objective is clear in its intent to categorise users based on their experience. It could be made measurable by specifying how the system will classify the levels of experience (e.g., number of levels, criteria for each level).
* **Profile Page Display:** The objective to display user profiles with specific attributes is specific. Measurability could be enhanced by including user interface responsiveness or load time expectations.
* **Workout and Nutrition Tracking:** The objectives related to tracking workouts and nutrition are detailed, outlining the features such as editable workout options, a dynamic workout database, and nutritional tracking through a meal database. To make these objectives measurable, one could specify the number of pre-made workouts available, the database's update frequency, or the accuracy of calorie calculations.

The analysis is well-modelled, offering a comprehensive understanding of the problem, user needs, and the rationale behind the proposed solution. It sets a solid foundation for the subsequent stages of design and development by clearly defining the project's scope and objectives grounded in real user requirements.

**The Analysis score: 8/9 Marks**

### 

### 

### **Documented Design**

The design documentation is thorough, including system design, data flow, ERD, and user interface, among others. The clear articulation of how the solution is structured across multiple components suggests a fully or nearly fully articulated design.

To improve this section, consider the following suggestions:

**Detailed System Flowcharts**: While the outline provides a high-level view, more detailed flow charts could enhance understanding. These should cover key processes in depth, such as user registration, workout and meal tracking, progress visualisation, and error handling.

**User Interface Design Enhancements:** The document mentions user interface design but does not delve into specifics. Including mock-ups or sketches of the UI, detailing the layout, navigation, and interaction design, would make the design phase more tangible. Additionally, considering user accessibility and ease of use in the interface design can ensure a wider audience can effectively use the application.

**Technical Specifications Clarity**: While hardware specifications are mentioned, clarifying the minimum and recommended system requirements for optimal performance can help potential users assess their capability to run the application effectively.

**Expand on Program Structure:** Further elaboration on the program's architecture, including how different modules interact, scalability considerations, and how the application ensures data security and privacy, would provide a more solid foundation for development.

**The Design Score: 10/12 Marks**

### **Technical Solution**

### Based on your statement “ I think I still have a few objectives left to achieve before the final deadline”, I assume not all requirements met at this stage hence the

### **Completeness of Solution Score : 11/15 Marks**

The techniques used in the project suggest a combination of GUI development with Tkinter, game programming concepts with Pygame, object-oriented programming (OOP) for database manipulation and display functionalities, and the use of APIs for exercise and nutrition data​​. This mix indicates a complex approach, incorporating various programming paradigms and technologies to create a comprehensive solution.

Given the diversity of techniques and the complexity of the implementation:

* Tkinter and Pygame are used for GUI and interactive elements, which require a solid understanding of user interface design and event-driven programming.
* OOP principles are applied to structure the application
* Utilisation of APIs for exercise and nutrition data demonstrates the ability to integrate external data sources and process complex data types

The technical solution's techniques used fall into Group A and excellent programming style

Make sure you add more comment in your code

**The Techniques Used Score : 27/27 Marks**

### **Testing**

The document shows evidence of thorough testing, including handling mistakes and using video links for demonstration. This approach indicates a robust testing phase that ensures the application's functionality and robustness. However, you need to add a table that has a test that you met all the requirements including evidence. You should use different types of test data such as normal, boundary and errournouse data.

**The testing score: 6/8**

### **Evaluation**

**Limited evaluation hence it score:1**