Reviewer Name: Eyad Ghanem Team Reviewed: Iron

- Does the design model (all diagrams) use the correct syntax? List what you think is wrong and suggest how to improve things.

- The use-case diagrams follow the standard UML syntax
- The class diagrams are well-structured, but they are missing some definitions such as race validation and payment handling.
- Sequence diagrams are good but I think error handling paths for failed registration or invalid licenses could be more detailed.
- Activity diagrams provide a logical workflow from my point of view but I think Display license invalid should have a final node as that does not continue on.

- Does the design model use the correct level of abstraction? If so, why? If not, where do you think the group could improve their design?

The project maintains a strong balance of abstraction, but the "Manage Account" use case combines multiple processes (profile update, payment, license renewal). Splitting it into smaller, distinct use cases would improve clarity.

- Are the diagrams consistent and form one design model?

The overall design is highly consistent, with logical transitions between different models. However, there are minor inconsistencies:

- Notifications and alerts are mentioned in use-case specifications but not in the class diagrams.
- Moderation for race feedback exists in use cases but not in the class diagram

---

- Does the design model (all diagrams together) fulfill the requirements from what was required for Deliverable 1/3? What requirements might be missing or wrong?

The core functionalities are well represented, but some aspects require additional depth:

- Payment processing details are vague. I think it will be better to show how are transactions validated
- Error-handling workflows for race registration (e.g., invalid license, full races) will be an add-on
- Organizer permissions for race modifications and cancellations are unclear.

- Based on the above, make suggestions on how to improve the overall design.

The overall design is excellent to me. But if we want to refine the design further, we can:

- Introduce explicit validation methods for racer eligibility within the Race and License classes.
- Add a Notification class to handle email alerts and race status updates.
- Expand RaceManager functionalities to include race modification and cancellation workflows.
- Clarify security protocols for login failures, retry limits, and authentication methods.
- Session expiration handling should be defined for login security.
- Detailed notification mechanisms should be explicitly modeled in diagrams.
- Alternative registration paths for racers who fail eligibility checks would improve usability.

- Does the Deliverable document contain all the information needed (based on the kickoff document)?

Yes, the deliverable document contain all information needed

- Your opinion about the design:

Your race management system is well-structured, highly detailed, and logically designed.
The diagrams clearly illustrate system interactions, with strong adherence to UML principles.

- Did you see anything in the design that might be valuable for your own team's design?
I like the personalized personas
