Identify exactly three aspects in Section 1 and Section 2 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection

A) **Weakness 1:** The goals are badly formulated. They are descriptive and force users to do certain actions, while they should be prescriptive and define what the stakeholders would like to achieve over world phenomena. This means they direct user actions rather than clearly defining desired outcomes and impacts, not fully capturing what the software should achieve. For example, it is said that: “Educators create code kata battles” while it should have been: “Enable educators to create and manage code kata battles”.

B) **Weakness 2:** Phenomena specification is insufficient and, in some cases, wrong.

   Wrong: For instance, looking at shared phenomena, the SP4 misunderstood the access process to tournaments and battles. Students join and eventually invite others to their team for each battle of a tournament, not for the entire challenge.

   Insufficient: the study of the end-of-the-game procedure is totally absent in the phenomena analysis. In SP9 it is pointed out the fact that CBK platform assigns badges to students, but this will become visible to everyone by looking into the user’s profile, not only by who earned them. Furthermore, CBK has to send a notification to all the students enrolled in a battle, with the final ranks. This procedure is also repeated for all the tournaments and should be included in the shared phenomena that are machine-controlled.

C) **Weakness 2:** The world phenomena WP1 is ambiguous because it states that educators code the automation scripts and create a test suite for the battle, but the specification states that the educator uses the CKB platform to upload test cases and build automation scripts. The problem is that they did not specify what is included in this phenomena, if they had clearly stated that the educator just thinks of the test suite, and there was a different shared phenomenon describing the upload of both the test cases and the automation scripts (which is not there) then it would have been ok to keep it as a world phenomenon.

Identify exactly three aspects in Section 3 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection.

A) **Weakness 1:** Many requirements are multi-faceted and would be clearer if divided into smaller, testable requirements, which would make the system easier to validate and maintain. An example of this is R12:

   “Educators can create a badge and a set of rules associated with that badge”
   It is poorly structured because it doesn’t define who allows who as it combines two distinct functionalities. It would be clearer if split into two separate requirements:
   - "The system allows the educator to create a badge."
   - "The system allows the educator to add a set of rules to assign the badge."

B) **Weakness 2:** There are some problems with the requirements about badges, in R13 The system lacks detail on who receives them, raising confusion about whether badges are assigned to individual students or entire teams. This information is crucial for correctly implementing badge distribution; while in R14 there is no information on where badges are displayed, specifically that they should appear on the student's profile. This detail is necessary to ensure users can view achievements as intended.

C) **Strength 1:** Despite the goals being somewhat vague, the connection between requirements, domain assumptions, and goals is well-defined, ensuring logical coherence and alignment (i.e., requirements and domain assumptions imply goals).

Identify exactly three aspects in Section 4 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection.

A) **Weakness 1:** The submission function is wrong:

fun submissions [g: Group]: set Submission{ (group: >g).dom }
The error outlined: “ Syntax error at line 29 column 51:There are 1 possible tokens that can appear here”
The correct version is:

```
fun submissions[g: Group]: set Submission {
        { s: Submission | s.group = g }
}
```

The corrected version filters Submission instances where each s has its group attribute equal to g, thus returning the correct set of submissions associated with the specified group.

B) **Weakness 2:** The modelers chose to focus on only one part of the system (ranking) without explaining why this was prioritized over other elements. The lack of reasoning behind this decision reduces the clarity and makes it harder to understand the significance of their verification efforts in the context of the whole system. Adding assertions to prevent counterexamples would also strengthen the model’s reliability.

C) **Strength 1:** The model effectively handles the verification process and addresses scenarios involving ties, demonstrating thoughtfulness in resolving potential ranking ambiguities. However, it could be improved by considering multiple battles and examples of dynamically changing group arrangements, an issue they addressed in Alloy.

Identify exactly three aspects in Section 1 and Section 2 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection

D) **Weakness 1:** The goals are badly formulated. They are descriptive and force users to do certain actions, while they should be prescriptive and define what the stakeholders would like to achieve over world phenomena. This means they direct user actions rather than clearly defining desired outcomes and impacts, not fully capturing what the software should achieve. For example, it is said that: “Educators create code kata battles” while it should have been: “Enable educators to create and manage code kata battles”.

E) **Weakness 2:** Phenomena specification is insufficient and, in some cases, wrong.

   Wrong: For instance, looking at shared phenomena, the SP4 misunderstood the access process to tournaments and battles. Students join and eventually invite others to their team for each battle of a tournament, not for the entire challenge.

   Insufficient: the study of the end-of-the-game procedure is totally absent in the phenomena analysis. In SP9 it is pointed out the fact that CBK platform assigns badges to students, but this will become visible to everyone by looking into the user’s profile, not only by who earned them. Furthermore, CBK has to send a notification to all the students enrolled in a battle, with the final ranks. This procedure is also repeated for all the tournaments and should be included in the shared phenomena that are machine-controlled.

F) **Weakness 2:** The world phenomena WP1 is ambiguous because it states that educators code the automation scripts and create a test suite for the battle, but the specification states that the educator uses the CKB platform to upload test cases and build automation scripts. The problem is that they did not specify what is included in this phenomena, if they had clearly stated that the educator just thinks of the test suite, and there was a different shared phenomenon describing the upload of both the test cases and the automation scripts (which is not there) then it would have been ok to keep it as a world phenomenon.

1.Weakness - Poorly Formulated Goals:

bad The goals they chose are badly formulated since they are descriptive and force users to do certain actions, while they should be prescriptive and define what the stakeholders would like to achieve over world phenomena. Also, the goals are not enough because they do not describe fully what the software must achieve.

- The goals are badly formulated. They are descriptive and force users to do certain actions, while they should be prescriptive and define what the stakeholders would like to achieve over world phenomena. This means they direct user actions rather than clearly defining desired outcomes and impacts, not fully capturing what the software should achieve. For example, it is said that: “Educators create code kata battles” while it should have been: “Enable educators to create and manage code kata battles”.

2.Weakness - Insufficient Phenomena Specification:

-> also SP4 totally wrong: students don’t create teams to join a tournament, but they join battles and for each of them create a team.
bad phenomena: everyone can have access to the results, badges, profiles…
This information is available for all students and educators subscribed to the CKB platform, that is, all users can see the list of ongoing tournaments as well as the corresponding tournament rank.
When an educator closes a tournament, as soon as the final tournament rank becomes available the CKB platform notifies all students involved in the tournament.

- *notification: user invited, deadline is expired,*

It could have been expanded way more instead of just fusing all of this information inside one phenomena.

- Phenomena specification is insufficient and, in some cases, wrong.
- Wrong: For instance, looking at shared phenomena, the SP4 misunderstood the access process to tournaments and battles. Students join and eventually invite others to their team for each battle of a tournament, not for the entire challenge.
- Insufficient: the study of the end-of-the-game procedure is totally absent in the phenomena analysis. In SP9 is pointed out the fact that CBK platform assigns badges to students, but this will become visible to everyone by looking into the user’s profile, not only by who earned them. Furthermore, CBK has to send a notification to all the students enrolled in a battle, with the final ranks. This procedure is also repeated for all the tournaments and should be included in the shared phenomena that are machine-controlled.

3.Weakness - Ambiguity in World Phenomena (WP1):

?? Another bad phenomena is WP1 because it might be shared since Educators creating a test suite for the battle sounds more like a world controlled shared phenomena because it directly affects the software (unless they mean they think of test cases)

- The phenomena described under WP1 are ambiguous, particularly regarding educator interactions with test suites. This seems more like a world-controlled, shared phenomenon as it impacts the system directly. This ambiguity could confuse distinguishing between shared and private interactions with the platform.
- The world phenomena WP1 is ambiguous because it states that educators code the automation scripts and create a test suite for the battle, but the specification states that the educator uses the CKB platform to upload test cases and build automation scripts, but they did not specify what is included in this phenomena. If they had clearly stated that the educator just thinks of the test suite, and there was a different shared phenomena describing the upload of both the test cases and the automation scripts (which is not there) then it would have been ok to keep it as a world phenomenon.

4.Strength - Comprehensive Scenarios: -> not so strong
good the scenarios are well formulated and cover all the possibility, also the class diagram seems to be well formulated (in realtà poi gli state diagram in confronto a quelli degli altri fanno un po schifo perchè sono semplicemente lineari quindi magari non scriviamolo).

- The scenarios are well-developed and cover all possibilities, which enhances the project's robustness by addressing a wide range of potential user interactions and situations, adding value to the requirement analysis phase.Either class diagram are well structured.

Identify exactly three aspects in Section 3 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection.

D) **Weakness 1:** Many requirements are multi-faceted and would be clearer if divided into smaller, testable requirements, which would make the system easier to validate and maintain. An example of this is R12:

“Educators can create a badge and a set of rules associated with that badge”
It is poorly structured because it doesn’t define who allows who as it combines two distinct functionalities. It would be clearer if split into two separate requirements:

- "The system allows the educator to create a badge."
- "The system allows the educator to add a set of rules to assign the badge."

E) Weakness 2: There are some problems with the requirements about badges, in R13 The system lacks detail on who receives them, raising confusion about whether badges are assigned to individual students or entire teams. This information is crucial for correctly implementing badge distribution; while in R14 there is no information on where badges are displayed, specifically that they should appear on the student's profile. This detail is necessary to ensure users can view achievements as intended.

F) Strength 1: Despite the goals being somewhat vague, the connection between requirements, domain assumptions, and goals is well-defined, ensuring logical coherence and alignment (i.e., requirements and domain assumptions imply goals).

1. R4 - Missing Subjective Entity:

requirement 4 -> The system allows who? -> the educators, missing the subjectiv
The system lacks clarity on who is allowed access, as it does not specify that educators are the intended users. This omission makes the requirement ambiguous and could lead to misunderstandings in user permissions.

2. R12 - Multiple Requirements in One:

requirement 12
Educators can create a badge and a set of rules associated with that badge
This is badly formulated and multiple requirement
->The system allows the educator to create a badge
->The system allows the educator to add a set of rules to assign the badge
“Educators can create a badge and a set of rules associated with that badge”
It is poorly structured because it doesn’t define who allows who as it combines two distinct functionalities. It would be clearer if split into two separate requirements:

- "The system allows the educator to create a badge."
- "The system allows the educator to add a set of rules to assign the badge."

3. R13 - Missing Recipient Information:
requirement 13 -> the system assigns the badges to whom? the entire team? -> to the single student (missing information)
- The system lacks detail on who receives badges, raising confusion about whether badges are assigned to individual students or entire teams. This information is crucial for correctly implementing badge distribution.

4. R14 - Unspecified Badge Visibility:
requirement 14 -> where are the badges visible? -> in the profile of the student
- There is no information on where badges are displayed, specifically that they should appear on the student's profile. This detail is necessary to ensure users can view achievements as intended.

5. USE CASE
PROBLEM IN USE CASES DIAGRAM:
- why in the use cases diagram github participates to battle begins?
- no predicate from student to take part in a tournament
- no reference to badge assigned (only creation)

- It's unclear why GitHub is involved in the "battle begins" use case, as its role isn't fully explained. Additionally, there's no indication that students participate in tournaments, nor is there any reference to badge assignment, only badge creation.

6. PERFORMANCE REQ:
performance requirements
not well defined because we can add
->The system must be available 7/7 24h
->The system must be performing even if there's a lot of concurrent usage
- Performance specifications are insufficiently defined; additional requirements could improve clarity:
  - "The system must be available 24/7."
  - "The system must perform reliably even under high concurrent usage."

7. CORRECT MAPPING:

Even though the goals are not well defined, the mapping between requirements and domain assumptions on those goals seems to be well defined and the logical implication (req + dom ⇒ goals)holds.

- Despite the goals being somewhat vague, the connection between requirements, domain assumptions, and goals is well-defined, ensuring logical coherence and alignment (i.e., requirements and domain assumptions imply goals).

8. USE CASES:

And the use cases are complete and well-formulated
Also sequence diagrams are very clear (although code evaluation could be explained better).

- The use case and sequence diagrams are well-constructed and comprehensive, providing clarity on user interactions and system flow, though a more detailed explanation of code evaluation processes could enhance the sequence diagrams further.

9. MULTIPLE REQUIREMENTS:

many requirements are multiple, it would have been better to split them because this way it's easier to test

- Many requirements are multi-faceted and would be clearer if divided into smaller, testable requirements, which would make the system easier to validate and maintain.

UNITI:

- Many requirements are multi-faceted and would be clearer if divided into smaller, testable requirements, which would make the system easier to validate and maintain. An example of this is R12:
  "Educators can create a badge and a set of rules associated with that badge"
  It is poorly structured because it doesn't define who allows who as it combines two distinct functionalities. It would be clearer if split into two separate requirements:
    - "The system allows the educator to create a badge."

- "The system allows the educator to add a set of rules to assign the badge."

- There are some problems with the requirements about badges, in R13 The system lacks detail on who receives them, raising confusion about whether badges are assigned to individual students or entire teams. This information is crucial for correctly implementing badge distribution; while in R14 there is no information on where badges are displayed, specifically that they should appear on the student's profile. This detail is necessary to ensure users can view achievements as intended.

- Despite the goals being somewhat vague, the connection between requirements, domain assumptions, and goals is well-defined, ensuring logical coherence and alignment (i.e., requirements and domain assumptions imply goals).

Identify exactly three aspects in Section 4 that represent either strengths or weaknesses (e.g., 1 strength and 2 weaknesses, or 3 weaknesses, etc.). Support each identified aspect with proper arguments (one or two sentences) motivating your selection.

D) Weakness 1: The submission function is wrong:
fun submissions [g: Group]: set Submission{ (group: >g).dom }
The error outlined: " Syntax error at line 29 column 51:There are 1 possible tokens that can appear here"
The correct version is:

```
fun submissions[g: Group]: set Submission {
        { s: Submission | s.group = g }
}
```

The corrected version filters Submission instances where each s has its group attribute equal to g, thus returning the correct set of submissions associated with the specified group.

E) **Weakness 2:** The modelers chose to focus on only one part of the system (ranking) without explaining why this was prioritized over other elements. The lack of reasoning behind this decision reduces the clarity and makes it harder to understand the significance of their verification efforts in the context of the whole system. Adding assertions to prevent counterexamples would also strengthen the model's reliability.

F) **Strength 1:** The model effectively handles the verification process and addresses scenarios involving ties, demonstrating thoughtfulness in resolving potential ranking ambiguities. However, it could be improved by considering multiple battles and examples of dynamically changing group arrangements, an issue they addressed in Alloy.

1.Weakness - Limited Scope of Modeling:

they have modeled only the ranking part (not the all environment -> Educator and badges part is missing)

- The current model focuses exclusively on the ranking aspect, omitting other critical components like the educator's role and badge assignments. This limits the model's usefulness for understanding the full system environment and its interactions.

  We could also add some assertion to be sure to not have counterexamples

2.Weakness - Unclear Participant Requirements:

not very clear when they model the participant of a tournament it is not necessary to have submitted

- The model is unclear regarding the requirements for tournament participation. It implies that submission is necessary to participate, although this may not be required, leading to potential misinterpretation of participation conditions.

3.Strength - Verification and Tie Handling:

ok the verification part -> smart when they model a tie
-> but they could at least consider more than one battle and within them the example of changing arrangements of the groups (they solved this issue in alloy)

- The model effectively handles the verification process and addresses scenarios involving ties, demonstrating thoughtfulness in resolving potential ranking ambiguities. However, it could be improved by considering multiple battles and examples of dynamically changing group arrangements, an issue they addressed in Alloy.

4.Weakness - Lack of Justification for Modeling Choices:
they did not explain anything about why what they verified is important, since they chose to only model 1 piece of the system they should have at least explained why they made this choice.

- The modelers chose to focus on only one part of the system (ranking) without explaining why this was prioritized over other elements. The lack of reasoning behind this decision reduces the clarity and makes it harder to understand the significance of their verification efforts in the context of the whole system. Adding assertions to prevent counterexamples would also strengthen the model's reliability.

5.Weakness 1 The submission function is wrong:

fun submissions [g: Group]: set Submission{ (group: >g).dom }
The error outlined: " Syntax error at line 29 column 51:There are 1 possible tokens that can appear here"
The correct version is:

```
fun submissions[g: Group]: set Submission {
        { s: Submission | s.group = g }
}
```

The corrected version filters Submission instances where each s has its group attribute equal to g, thus returning the correct set of submissions associated with the specified group.