

Ci-dessous l'énoncé :

```
## Use case
**Product Goal :** A simple RPG character creator and idle game to test
the market and the concept
**A character has :**
- Skill points
- Health
- Attack
- Defense
- Magik
**All characters start at level 1 with :**
- Skill points : 12
- Health : 10
- Attack : 0
- Defense : 0
- Magik : 0
**The player can distribute Skill Points to Health, Attack, Defense and
Magik following the rules :**
- Health : increasing 1 Health Point costs 1 Skill Point regardless the
Health Point amount
- Attack, Defense, Magik : increasing 1 Skill Point costs Skill's
amount divided by 5 Skill Point, rounded at superior
  _**Example :**_
  - Health : 44, costs 1 Skill Point to increase to 45
  - Attack : 3, costs 1 Skill Point to increase to 4
    _3 / 5 = 0.6 => 1 Skill Points_
  - Defense : 9, costs 2 Skill Points to increase to 10
    _9 / 5 = 1.8 => 2 Skill Points_
  - Magik : 32, costs 7 Skill Points to increase to 33
    _32 / 5 = 6.4 => 7 Skill Points_
- Skill Points can't be negative (minimum 0)
- Until validation a Skill Point assigned to a skill can be retrieved
and reassigned.
- After validation, a Skill Point can't be retrieved
**After each fight :**
- Health Points are restored
- when a character loses : it can't fight for 1 hour
- when a character wins : it gains 1 Skill Point
**Character rank is :**
- All character starts at rank : 1
- rank is increased by 1 when a character wins
- rank is decreased by 1 when a character loses
- rank can't drop below 1
**Fight are random turn based :**
- The player's character is the first character, the opponent is the
second
- Each turn, both characters roll a dice with as many faces as the
Attack's Skill Point amount, it's the Attack's value
  _**Example :**_
  - Gaston has 10 Skill Points in Attack
```

- he launches a 1D10 dice (a.k.a. a dice with 10 faces : it results a number between 1 and 10)
- Mathilda has 5 Skill Points in Attack
 - she launches a 1D5 dice (a.k.a. a dice with 5 faces : it results a number between 1 and 5)
- *1D0 dice doesn't exist, result is always 0*
- *The Attack's value are compared with Defense's Skill Point amount, if the difference is :*
 - positive : Attack succeed
 - zero or negative : Attack failed
- *When an Attack succeeds the difference is subtracted from the opponent's Health Point*
- *If the difference equals Magik's Skill Point amount, this value is added to the difference*
- *****Exemple :*****
 - Player with Gaston launches the fight : Gaston will always play first
 - Gaston has 10 Skill Points in Attack, he launches 1D10 dice and obtains 10.
 - Gaston has 7 Skill Points in Magik.
 - Mathilda has 3 Skill Points in Defense, difference is $10 - 3 = 7$, same value as Gaston's Magik Skill Points
 - Mathilda receives $7 + 7 = 14$ damages, if she has 24 Health Skill Points : $24 - 14 = 10$ remains
- *Until a character's Health Point reaches 0 (or less), the fight continues.*
- *When a character has no more Health Point, fight is instantly finished.*
- *****A Player can only interact with his/her characters :*****
 - *He/She can sign up and create a character*
 - *He/She can sign in and :*
 - retrieve the list of his/her characters
 - consult a character's details
 - create a new character (maximum 10 characters per player)
 - update a character (only if some Skill Points are available)
 - delete a character
 - retrieve the list of fights for a character with the result (won/loose)
 - *launch a new fight in the lobby*
- *He/She can sign out*
- *****Launching a fight*****
 - *Player chooses an existing character in his/her pool*
 - *Enter the lobby to fight*
 - *An opponent is automatically chosen following this rules :*
 - take the closest opponent based on rank value
 - the opponent has to be free (it must not have fight in the past hour)
 - if several opponents match, take the opponent with the smallest number of fights with the character
 - if several opponents match, take a random opponent within the list
 - *Player is informed by a report with each turn's details :*
 - Turn count (start at 1)
 - Attack's value for both characters

- Health point subtracted for both characters
- Fight status (won/loose)

Technical Requirements

- You can use any programming language but nothing too exotic
- You can use any framework but nothing too exotic
- You should provide code as clean as possible according to best practices focusing on maintainability and evolutivity
 - Automation Tests (good strategy)
 - "Clean Code"
 - Decoupling
 - SOLID principles
 - Good architecture
 - Documentation (if we can't launch it, it's useless)
 - Tooling to ensure quality
- No need to deploy it publicly, dev environment on public repository is enough

Notes

- No need to deliver all the features if it's too much but prioritise according to the product goal above
- All requierements above are written on purpose, you can improve/rewrite them (User Story, Acceptance Criteria, Gherkin...)