Cycling coursework – info

# Overall task

Making a back end to a cycling system. The system needs to be able to manage riders and the teams. It also needs to be able to create races with stages within them as well as the parts of the stages that can earn the riders points. The race results also need to be handled and then record them (may not need to process the record and output them.

The source code needs to be commentated and include assertions.

# Riders and teams

## Riders

The system needs to be able to add and remove riders as well as changing the team that they are on. Each rider needs an teamID, name and the year of their birth. When a rider is removed, all information about the rider needs to be removed including their results from the system and the race results need to be updated.

## Teams

System needs to manage the teams within the races, teams need to be created and deleted. To create a team, the team needs a name and description and the teamID is then created as part of the program.

# Staged race

To create a race, the user needs to input all the information needed for the stages. The user also need to input the name of the race and a description of the race and then return the id of the race. There are three winners for each race in the following categories.

## General classification

Each rider gets a time for each stage of the race. The winner is the rider who has the smallest amount of time after all of the stages has been added together.

## Points (Sprinter’s) Classification

Points are awarded to the riders at the end of each race depending on place they finish. Each stage needs to be classified into 1 of 3 stages: flat (mainly 4 and some 3 hight mountains), medium (2 or 3 hight mountains), high (1 or HC, Hors catégorie, mountains). Within each stage there can be an intermediate sprint and whoever finishes the sprint first gets the most points. The rider that has the most points at the end of all the stages, wins the classifications. Graphical user interface, application

Description automatically generated

## Mountain classification

Graphical user interface, application

Description automatically generatedWithin each stage if there is a mountain, points are awarded based off whoever reaches the summit first. After all the stages are complete, the player with the most points is the KOM (King of the Mountain).

## Information needed to create a stage

To create a stage the following information is needed:

* raceId (the race that the stage is a part of)
* name of the stage
* the description of the stage
* The length
* Local date and time
* The start time
* The type of stage (flat, medium or high)

There are different segments that can be added to a stage. There segments can also be removed from the stage using the segment id. Once the stage and all the segments are inputted then the stage can be concludes and should be “waiting for results”.

### Information to add a climb to the stage

To add a climb to a stage the following information is needed:

* stagid (the stage that climb is in)
* location -km where the sprint finishes
* type of climb (4, 3, 2, 1 or HC)
* Average gradient
* Length

Id of the segments needs to be returned

### Information to add a intermediate Sprint to the stage

To add an intermediate to a stage the following information is needed:

* Stageid
* Location – Km location where the sprint finishes

Id of the segments needs to be returned

# NEED TO DO

* A cover page
* A printout