“Tilmann’s Cijfer” generator – design

Introduction

Giving marks to students cost a lot of time and pain. For this a automatic solution is needed. Therefore the overall Software architect (Tilmann K.) has designed a solution for the ROS2 architecture.

The solution design has following ROS2 nodes

- “tentamen” result generator
- final “cijfer” determinator
- cijfer calculator
- “herkansing “cijfer” determinator
- herkansing scheduler

The interfaces of the design are:
Msg:

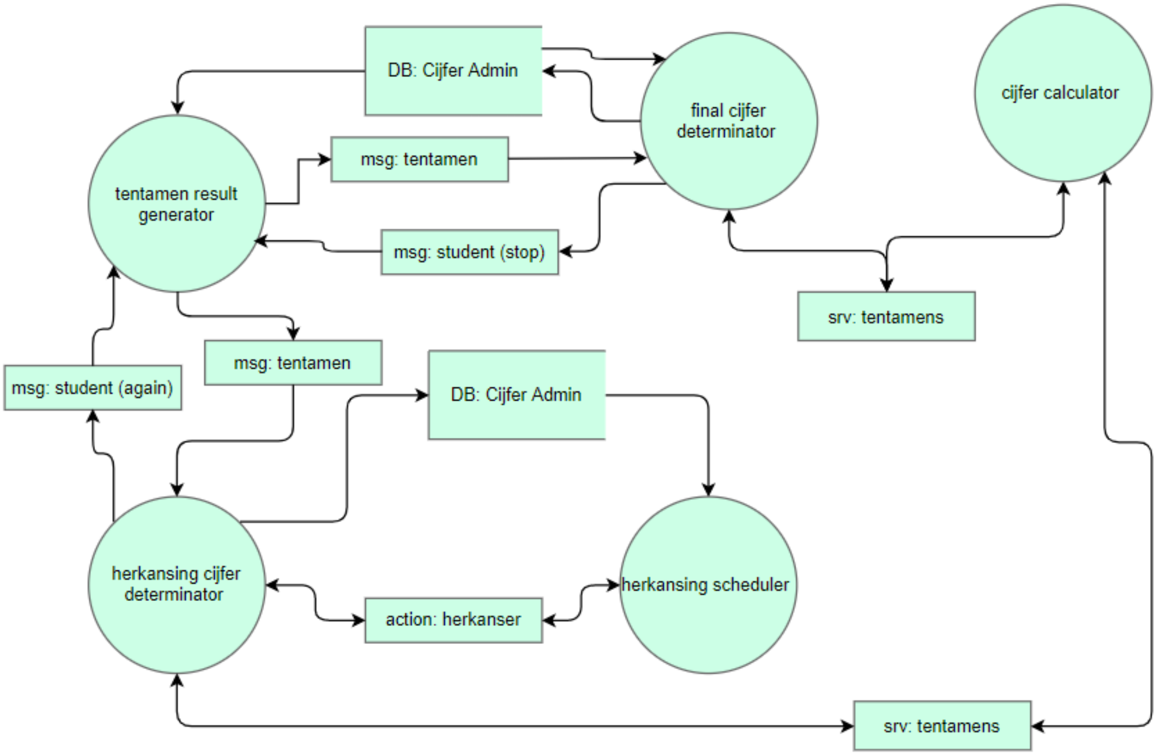
- tentamen
- student

Srv:

- tentamens

Action:

- herkanser



Nodes

“tentamen” result generator

The tentamen result generator node collects from a Database (file) all student/course combinations for which ”tentamen” results need to be generated. These are the student/course combinations for which the final course result does not exist. After the collection the node creates and broadcastes randomly an exam mark (between 10 and 100) for a random student/course combination (meaning e.g. every two seconds a result is published for a random student/course combination. The node can receives a messages that will ask to add or to remove a student/course combination to the random generation process

final “cijfer” determinator

The receives the tentamen results from the result generator. After it has collected enough results (enough: = the collection number that was retrieved from the database) it sends a request to the “cijfer” calculator node. After it got the final cijfer back it inserts the final “cijfer” in the database and sends a message to the cijfer generator to stop the generation for this student/course combination.

cijfer calculator

The cijfer calcualor node can receives requests to calculate a final result that is returned. This is done by calculating the average of several exam results. The student Wessel gets always a bonus of 10 points. The final result has to be a number between 10 and 100.

herkansing scheduler

From time to time (regularly) the node collects all students that failed a course. These are the students that have a final course result between 10 and 54. For all these students it requests to start a new cijfer determination.

herkansing “cijfer” determinator

This node executes the a new final cijfer determination for a student/course combination by receiving again randon exams results from the result generator. For this the random generation has to be activated again by sending a message to the result generator node. After receiving again enough results the exams are sent to the cijfer calculator to determine a new final result. The new final result is added to the database (note: the old one should not be overwritten).

Interfaces

Msg: student

Contains a student/course combination

Msg: tentamen

Contains a student/course combination with additional a “tentamen cijfer”

All should contain a ROS2 standaard timestamp.

Srv: tentamens

Sends a student/course combination with a list of tentamen cijfers and returns a final result of a student/course combination.

Action: herkanser
Sends a student/course combination

Database

The database (file) contains the following table (not normalized)
Student name , course , number of exams , final result , timestamp