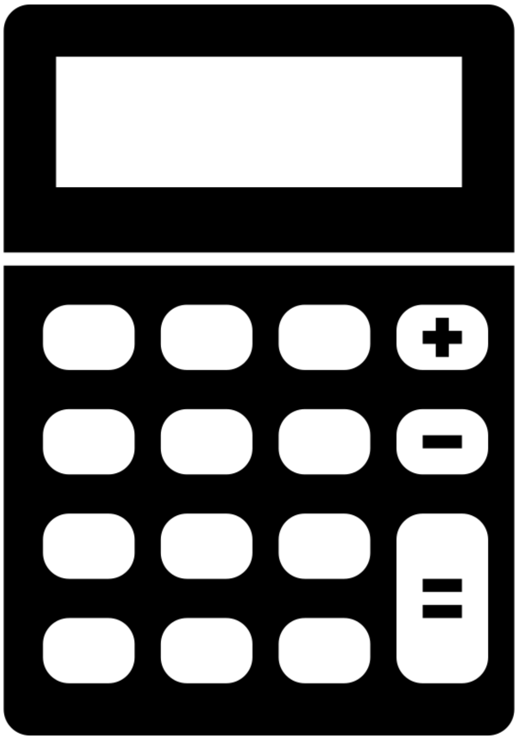
Node and Integration Tests for Tilmann’s cijfer generator

Group 4: Rik van Velzen, Melissa van Leeuwen, Burhan Topaloglu



This document details the testing we have performed for our package solution.

CONTENTS

Testing Method

Unit test

Test different scenarios for the units themselves, try all inputs and check the output against expected outputs for each unit.
The Unit test is successful when every unit is tested on its own.

System test (includes integration test)

Tests the whole system. Try all inputs and check against Expected outputs.
A system test on a complete system where every unit is dependent on another doubles as an integration test, as this system should only work if all parts properly interact.
The system test and thus also the integration test is successful when the entire system gives expected output against input.

Acceptance test

“Acceptance tests are formal tests that verify if a system satisfies business requirements”
Usually achieved through comparing the resulting system to the user stories that were created during the debriefing phase. During this testing, one is to act like a typical user.
The acceptance test is successful if this typical user does not encounter any problems, and all essential customer/end-user demands are met(usually measured by user stories).

ROS built-in testing

To run these tests we can manually perform inputs on the parameters.
ROS also offers ways to test nodes in the documentation:

Write cpp test:
https://docs.ros.org/en/jazzy/Tutorials/Intermediate/Testing/Cpp.html
(Though personally this guide was way better than the docs)
https://automaticaddison.com/how-to-create-unit-tests-with-gtest-ros-2-jazzy/
Running the tests:
https://docs.ros.org/en/jazzy/Tutorials/Intermediate/Testing/CLI.html
We won't be using the ROS Build Farm this time..

Node/Unit Tests

Does every Unit work?

Retake Scheduler

Scenarios:

1. One student received a passed grade for the course
   Expected outcome:
    - The student is not added to the retakeQueue.
    - scheduledRetakes remains empty.

   Input data:
   student_id = 2
   course_id = 1
   Number_of_exams = 3
   final_grade = 75.00
   Results:
   retakeQueue and scheduledRetakes stay empty for a passed grade.
2. One student received a failing grade for a course
   Expected outcome:
    - The student is added to retakeQueue.
    - scheduledRetakes contains the (student_id, course_id) combination.
    - The queue size is 1.

   Input data:
   student_id = 4
   course_id = 4
   Number_of_exams = 3
   final_grade = 40.00
   Results:
   Size of retakeQueue is 1, scheduledRetakes and retakeQueue are not empty.
3. Two students received a failing grade and one student a passed grade
   Expected outcome:
    - Only the two failing students are added to retakeQueue.
    - scheduledRetakes only includes the failing combinations.
    - Queue size is 2.

   Input data:
   student_id = 5
   course_id = 1
   Number_of_exams = 3
   final_grade = 45.00

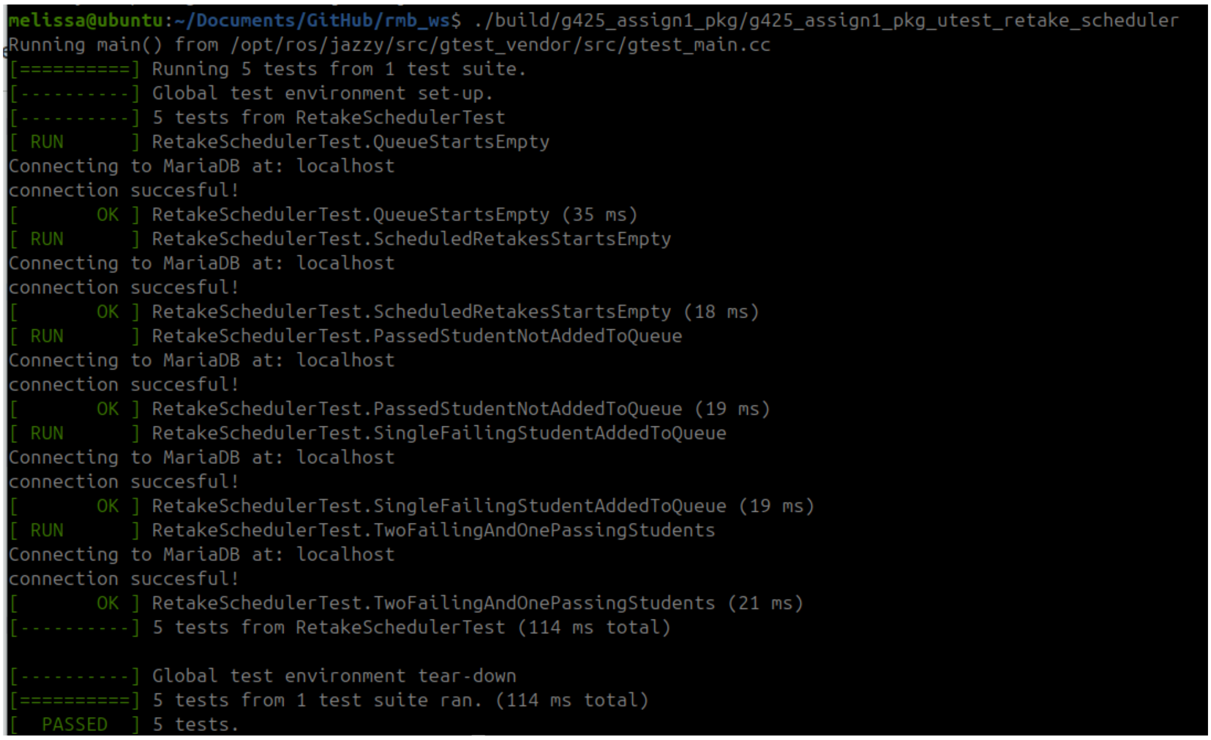
   student_id = 6
   course_id = 2
   Number_of_exams = 3
   final_grade = 30.00

student_id = 3
course_id = 3
Number_of_exams = 4
final_grade = 80.00

Results:

Size of retakeQueue and scheduledRetakes is 2. Only the two failing combinations (student_id and course_id) are added to scheduledRetakes. The passed combination is not added to scheduledRetakes.

Results:



```
melissa@ubuntu:~/Documents/GitHub/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_retake_scheduler
Running main() from /opt/ros/jazzy/src/gtest_vendor/src/gtest_main.cc
[==========] Running 5 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 5 tests from RetakeSchedulerTest
[ RUN      ] RetakeSchedulerTest.QueueStartsEmpty
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] RetakeSchedulerTest.QueueStartsEmpty (35 ms)
[ RUN      ] RetakeSchedulerTest.ScheduledRetakesStartsEmpty
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] RetakeSchedulerTest.ScheduledRetakesStartsEmpty (18 ms)
[ RUN      ] RetakeSchedulerTest.PassedStudentNotAddedToQueue
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] RetakeSchedulerTest.PassedStudentNotAddedToQueue (19 ms)
[ RUN      ] RetakeSchedulerTest.SingleFailingStudentAddedToQueue
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] RetakeSchedulerTest.SingleFailingStudentAddedToQueue (19 ms)
[ RUN      ] RetakeSchedulerTest.TwoFailingAndOnePassingStudents
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] RetakeSchedulerTest.TwoFailingAndOnePassingStudents (21 ms)
[----------] 5 tests from RetakeSchedulerTest (114 ms total)

[----------] Global test environment tear-down
[==========] 5 tests from 1 test suite ran. (114 ms total)
[  PASSED  ] 5 tests.
```

Retake Grade Determinator

Scenarios:

1. Test the examResultsCallback() function by sending exam grades for a student.
   Expected outcome:
   - The received exam grades are added to collected_grades_.
   - The number of collected grades matches number_of_exams.

   Input data:

   student_id = 2
   course_id = 2
   number_of_exams = 3
   Grades sent via callback = 80.0, 80.0, 80.0

   Results:

   The size of collected_grades_ is 3 and all grades are equal to 80.

1. Test the execute() function with a student whose average grades are above the retake threshold.
   Expected outcome:
   - The student passes the retake.

   Input data:

   student_id = 1
   course_id = 1
   number_of_exams = 3
   Grades (mocked in test) = [70, 75, 80]
   retake_grade_ = 55.0

   Results:
   Succeeded_ = true
   Aborted_ = false
   result_->message = “Retake passed”

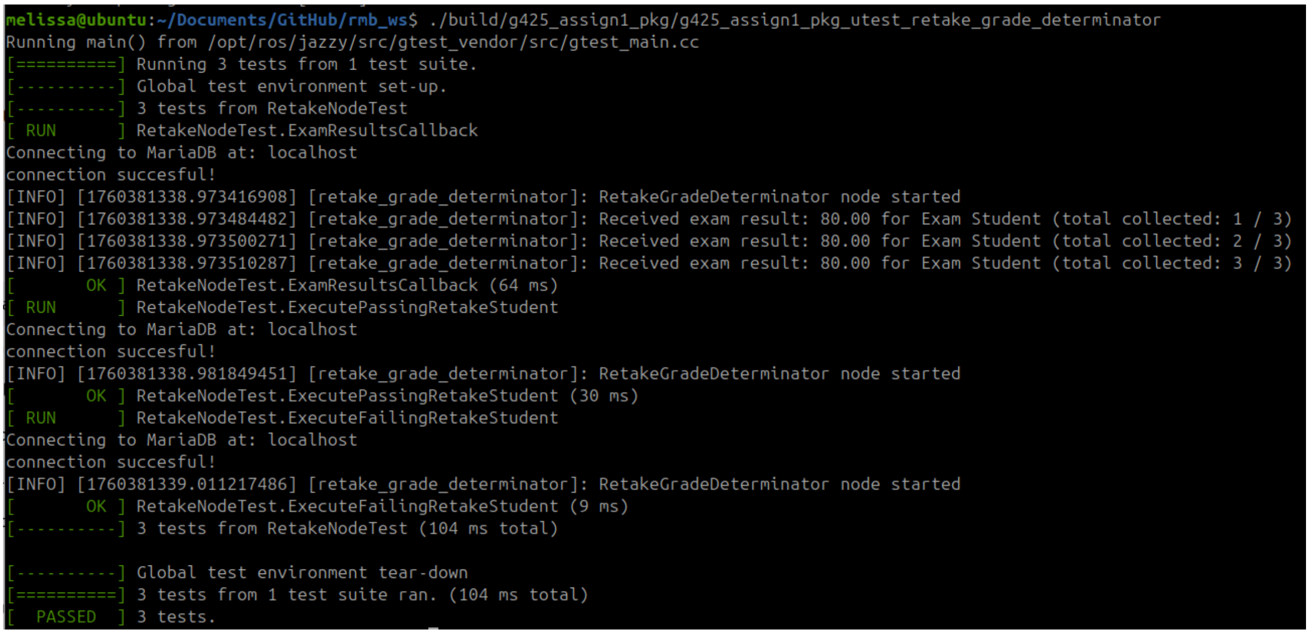
2. Test the execute() function with a student whose average grades are below the retake threshold.
   Expected outcome:
   - The student fails the retake.

   Input data:

   student_id = 3
   course_id = 1
   number_of_exams = 3
   Grades (mocked in test) = [70, 75, 80]
   retake_grade_ = 90.0

   Results:
   Succeeded_ = false
   Aborted_ = true
   result_->message = “Retake failed”

Results:



```
melissa@ubuntu:~/Documents/GitHub/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_retake_grade_determinator
Running main() from /opt/ros/jazzy/src/gtest_vendor/src/gtest_main.cc
[==========] Running 3 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 3 tests from RetakeNodeTest
[ RUN      ] RetakeNodeTest.ExamResultsCallback
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760381338.973416908] [retake_grade_determinator]: RetakeGradeDeterminator node started
[INFO] [1760381338.973484482] [retake_grade_determinator]: Received exam result: 80.00 for Exam Student (total collected: 1 / 3)
[INFO] [1760381338.973500271] [retake_grade_determinator]: Received exam result: 80.00 for Exam Student (total collected: 2 / 3)
[INFO] [1760381338.973510287] [retake_grade_determinator]: Received exam result: 80.00 for Exam Student (total collected: 3 / 3)
[       OK ] RetakeNodeTest.ExamResultsCallback (64 ms)
[ RUN      ] RetakeNodeTest.ExecutePassingRetakeStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760381338.981849451] [retake_grade_determinator]: RetakeGradeDeterminator node started
[       OK ] RetakeNodeTest.ExecutePassingRetakeStudent (30 ms)
[ RUN      ] RetakeNodeTest.ExecuteFailingRetakeStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760381339.011217486] [retake_grade_determinator]: RetakeGradeDeterminator node started
[       OK ] RetakeNodeTest.ExecuteFailingRetakeStudent (9 ms)
[----------] 3 tests from RetakeNodeTest (104 ms total)

[----------] Global test environment tear-down
[==========] 3 tests from 1 test suite ran. (104 ms total)
[  PASSED  ] 3 tests.
```

Result Generator

Scenarios:

1. Add one test student-course combination.
   Expected outcome:
   - One student-course combination is added to the vector of student msgs.

   Input data:

   student.student_fullname = Test Student
   student.student_id = 12345
   student.course_name = Test Course
   student.course_id = 101

   Results:

   Student was successfully added to the student vector.
2. Add the same test student-course combination again.
   Expected outcome:
   - Node gives a warning that the student already exists in the vector, student not added.

   Input data:

   student.student_fullname = Test Student
   student.student_id = 12345
   student.course_name = Test Course
   student.course_id = 101

   Results:

   Student wasn't added to the student vector, because the student already exists.
3. Remove the test student-course combination from the students vector.
   Expected outcome:
   - The test student-course combination is removed from the vector of student msgs.

   Input data:

   student.student_fullname = Test Student
   student.student_id = 12345
   student.course_name = Test Course
   student.course_id = 101

   Results:

   Student was successfully removed from the student vector.
4. Try to remove the test student-course combination from the students vector again.
   Expected outcome:
   - Node gives a warning that the test student does not exist in the vector of student msgs.

   Input data:

   student.student_fullname = Test Student
   student.student_id = 12345
   student.course_name = Test Course
   student.course_id = 101

   Results:

   Student not found, couldn't remove student.

5. Publish a random result without any students.

   Expected outcome:

   - The node gives a warning that there are no students available.

   Input data:

   No input data.

   Results:

   Didn't publish a result because there are no students.

6. Publish a random result with students.

   Expected outcome:

   - The node publishes a result with corresponding student-course combination.

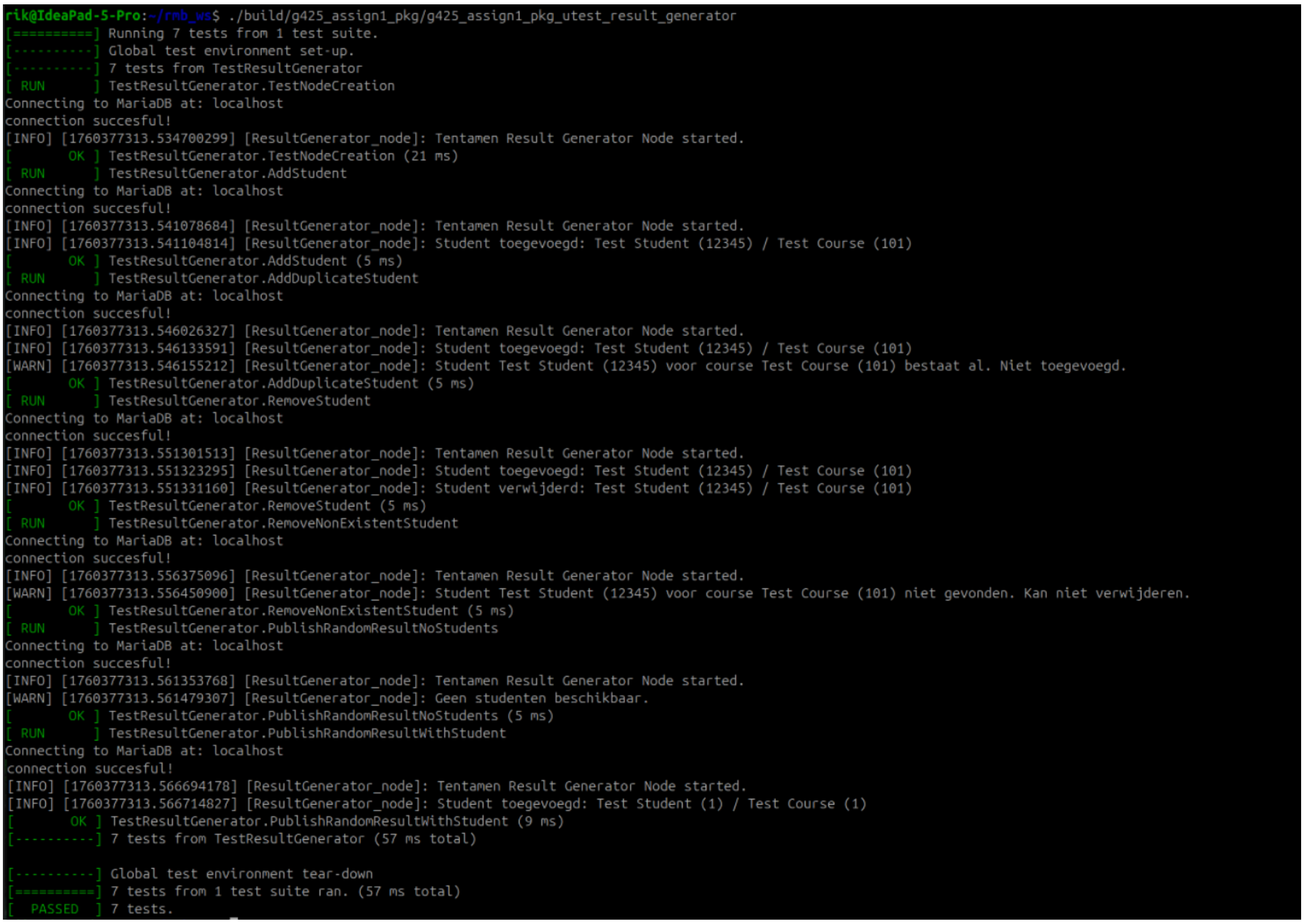
   Input data:

   student.student_fullname = Test Student
   student.student_id = 1
   student.course_name = Test Course
   student.course_id = 1

   Results:

   Published result with matching student.

Results:



```
rik@IdeaPad-5-Pro:~/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_result_generator
[==========] Running 7 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 7 tests from TestResultGenerator
[ RUN      ] TestResultGenerator.TestNodeCreation
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.534700299] [ResultGenerator_node]: Tentamen Result Generator Node started.
[       OK ] TestResultGenerator.TestNodeCreation (21 ms)
[ RUN      ] TestResultGenerator.AddStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.541078684] [ResultGenerator_node]: Tentamen Result Generator Node started.
[INFO] [1760377313.541104814] [ResultGenerator_node]: Student toegevoegd: Test Student (12345) / Test Course (101)
[       OK ] TestResultGenerator.AddStudent (5 ms)
[ RUN      ] TestResultGenerator.AddDuplicateStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.546026327] [ResultGenerator_node]: Tentamen Result Generator Node started.
[INFO] [1760377313.546133591] [ResultGenerator_node]: Student toegevoegd: Test Student (12345) / Test Course (101)
[WARN] [1760377313.546155212] [ResultGenerator_node]: Student Test Student (12345) voor course Test Course (101) bestaat al. Niet toegevoegd.
[       OK ] TestResultGenerator.AddDuplicateStudent (5 ms)
[ RUN      ] TestResultGenerator.RemoveStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.551301513] [ResultGenerator_node]: Tentamen Result Generator Node started.
[INFO] [1760377313.551323295] [ResultGenerator_node]: Student toegevoegd: Test Student (12345) / Test Course (101)
[INFO] [1760377313.551331160] [ResultGenerator_node]: Student verwijderd: Test Student (12345) / Test Course (101)
[       OK ] TestResultGenerator.RemoveStudent (5 ms)
[ RUN      ] TestResultGenerator.RemoveNonExistentStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.556375096] [ResultGenerator_node]: Tentamen Result Generator Node started.
[WARN] [1760377313.556450900] [ResultGenerator_node]: Student Test Student (12345) voor course Test Course (101) niet gevonden. Kan niet verwijderen.
[       OK ] TestResultGenerator.RemoveNonExistentStudent (5 ms)
[ RUN      ] TestResultGenerator.PublishRandomResultNoStudents
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.561353768] [ResultGenerator_node]: Tentamen Result Generator Node started.
[WARN] [1760377313.561479307] [ResultGenerator_node]: Geen studenten beschikbaar.
[       OK ] TestResultGenerator.PublishRandomResultNoStudents (5 ms)
[ RUN      ] TestResultGenerator.PublishRandomResultWithStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760377313.566694178] [ResultGenerator_node]: Tentamen Result Generator Node started.
[INFO] [1760377313.566714827] [ResultGenerator_node]: Student toegevoegd: Test Student (1) / Test Course (1)
[       OK ] TestResultGenerator.PublishRandomResultWithStudent (9 ms)
[----------] 7 tests from TestResultGenerator (57 ms total)

[----------] Global test environment tear-down
[==========] 7 tests from 1 test suite ran. (57 ms total)
[  PASSED  ] 7 tests.
```

Final Grade Determinator

Scenarios:

1. Receive a grade for an unknown student.
   Expected outcome:
   - Grade for unknown student, ignored the result.

   Input data:
   student_id = 9999
   course_id = 8888
   student_fullname = Unknown Student
   course_name = Unknown course
   Exam_grade = 85.0
   Results:
   Grade is received, but dropped, because there was no matching student.
2. Receive a grade for a known student.
   Expected outcome:
   - The grade is stored in a map with student-course information.

   Input data:
   student_id = 1
   course_id = 1
   student_fullname = Test Student
   course_name = Test course
   Number_of_grades = 3
   exam_grade = 90.0
   Results:
   One grade of 90.0 is received and added to the student.
3. Retrieve students from the database.
   Expected outcome:
   - Student is pulled from the database and added to the vector of student msgs.

   Input data:
   Database data or:
   student_id = 1
   course_id = 1
   student_fullname = Test Student
   course_name = Test course
   Number_of_grades = 3
   Results:
   One or more students are added to the student vector.
4. The student-course combination vector is empty.
   Expected outcome:
   - Check database again, when empty do nothing, with a new student add it to the vector.

   Input data:
   student_id = 1
   course_id = 1
   student_fullname = Test Student
   course_name = Test course

Number_of_grades = 3

Results:

When the student vector is empty one or more students are added to the student vector.

5. The grade calculator node is offline.

   Expected outcome:

   - Try a couple of times to connect to the node, when not connected, drop the student.

   Input data:

   student_id = 1
   course_id = 1
   student_fullname = Test Student
   course_name = Test course
   number_of_grades = 3
   grades = [85.0, 90.0, 78.0]

   Results:

   The grade calculator node is offline, and the student is dropped.

6. Sending grades to the Grade calculator node.

   Expected outcome:

   - Sending a couple of grades to the calculator, and confirming how many grades are sent.

   Input data:

   student_id = 1
   course_id = 1
   student_fullname = Test Student
   course_name = Test course
   number_of_grades = 3
   grades = [85.0, 90.0, 78.0]

   Results:

   3 grades are sent to the node.

7. Receiving grades for multiple students.

   Expected outcome:

   - All grades are recorded for the matching student-course combination

   Input data:

   3 students with every student i++
   student_id = i
   course_id = i
   student_fullname = Student i
   course_name = Course i
   number_of_grades = 2

   Results:

   For each student 2 grades are recorded, 3 students in total.

8. Data getting cleared.

   Expected outcome:

   - All data is cleared from memory.

   Input data:

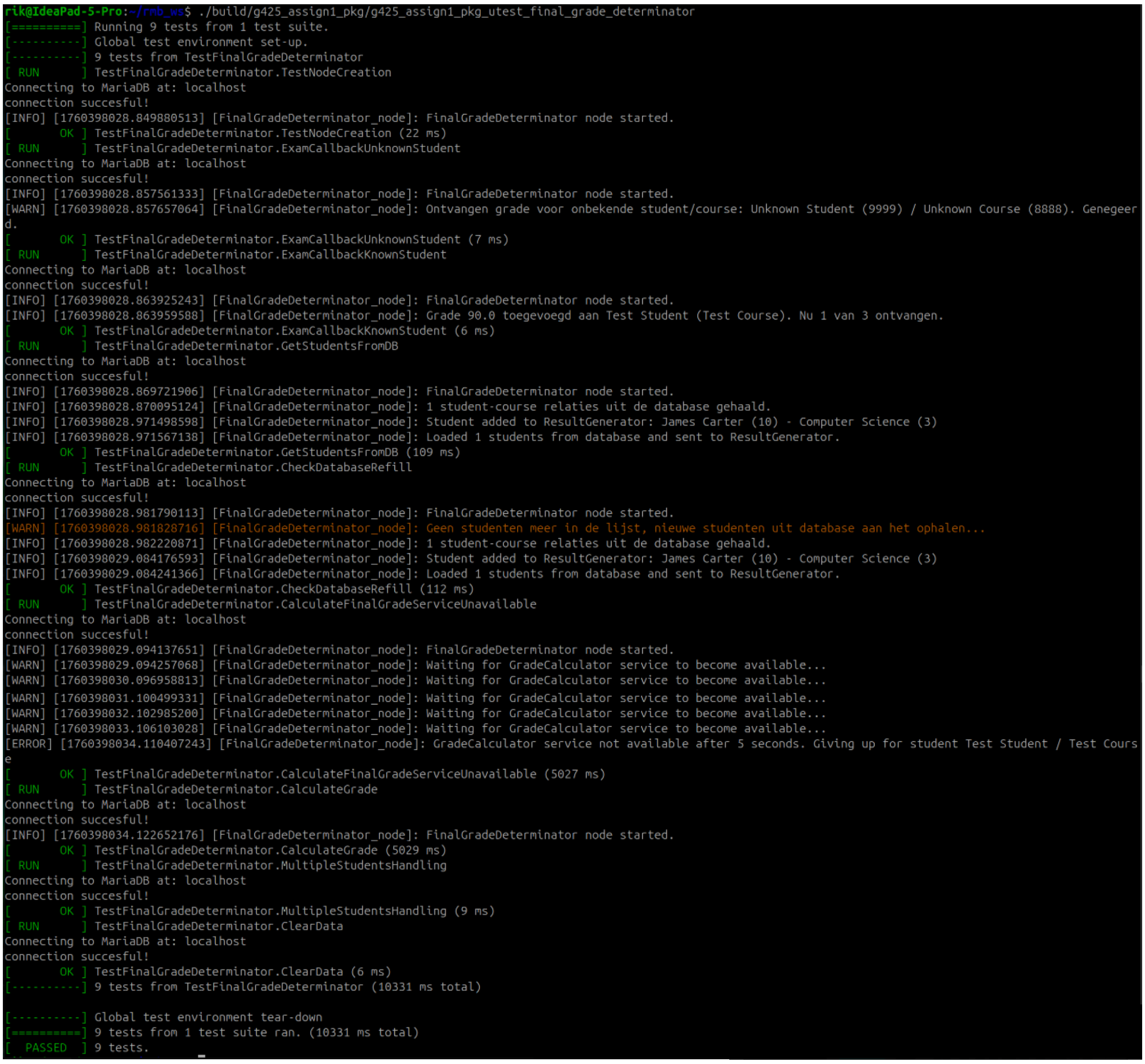
   student_id = 1
   course_id = 1

student_fullname = Test Student
course_name = Test course
number_of_grades = 3

Results:

All data is cleared.

Results:



```
rik@IdeaPad-5-Pro:~/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_final_grade_determinator
[==========] Running 9 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 9 tests from TestFinalGradeDeterminator
[ RUN      ] TestFinalGradeDeterminator.TestNodeCreation
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398028.849880513] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[       OK ] TestFinalGradeDeterminator.TestNodeCreation (22 ms)
[ RUN      ] TestFinalGradeDeterminator.ExamCallbackUnknownStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398028.857561333] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[WARN] [1760398028.857657064] [FinalGradeDeterminator_node]: Ontvangen grade voor onbekende student/course: Unknown Student (9999) / Unknown Course (8888). Genegeer
d.
[       OK ] TestFinalGradeDeterminator.ExamCallbackUnknownStudent (7 ms)
[ RUN      ] TestFinalGradeDeterminator.ExamCallbackKnownStudent
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398028.863925243] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[INFO] [1760398028.863959588] [FinalGradeDeterminator_node]: Grade 90.0 toegevoegd aan Test Student (Test Course). Nu 1 van 3 ontvangen.
[       OK ] TestFinalGradeDeterminator.ExamCallbackKnownStudent (6 ms)
[ RUN      ] TestFinalGradeDeterminator.GetStudentsFromDB
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398028.869721906] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[INFO] [1760398028.870095124] [FinalGradeDeterminator_node]: 1 student-course relaties uit de database gehaald.
[INFO] [1760398028.971498598] [FinalGradeDeterminator_node]: Student added to ResultGenerator: James Carter (10) - Computer Science (3)
[INFO] [1760398028.971567138] [FinalGradeDeterminator_node]: Loaded 1 students from database and sent to ResultGenerator.
[       OK ] TestFinalGradeDeterminator.GetStudentsFromDB (109 ms)
[ RUN      ] TestFinalGradeDeterminator.CheckDatabaseRefill
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398028.981790113] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[WARN] [1760398028.981828716] [FinalGradeDeterminator_node]: Geen studenten meer in de lijst, nieuwe studenten uit database aan het ophalen...
[INFO] [1760398028.982220871] [FinalGradeDeterminator_node]: 1 student-course relaties uit de database gehaald.
[INFO] [1760398029.084176593] [FinalGradeDeterminator_node]: Student added to ResultGenerator: James Carter (10) - Computer Science (3)
[INFO] [1760398029.084241366] [FinalGradeDeterminator_node]: Loaded 1 students from database and sent to ResultGenerator.
[       OK ] TestFinalGradeDeterminator.CheckDatabaseRefill (112 ms)
[ RUN      ] TestFinalGradeDeterminator.CalculateFinalGradeServiceUnavailable
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398029.094137651] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[WARN] [1760398029.094257068] [FinalGradeDeterminator_node]: Waiting for GradeCalculator service to become available...
[WARN] [1760398030.096958813] [FinalGradeDeterminator_node]: Waiting for GradeCalculator service to become available...
[WARN] [1760398031.100499331] [FinalGradeDeterminator_node]: Waiting for GradeCalculator service to become available...
[WARN] [1760398032.102985200] [FinalGradeDeterminator_node]: Waiting for GradeCalculator service to become available...
[WARN] [1760398033.106103028] [FinalGradeDeterminator_node]: Waiting for GradeCalculator service to become available...
[ERROR] [1760398034.110407243] [FinalGradeDeterminator_node]: GradeCalculator service not available after 5 seconds. Giving up for student Test Student / Test Cours
e
[       OK ] TestFinalGradeDeterminator.CalculateFinalGradeServiceUnavailable (5027 ms)
[ RUN      ] TestFinalGradeDeterminator.CalculateGrade
Connecting to MariaDB at: localhost
connection succesful!
[INFO] [1760398034.122652176] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[       OK ] TestFinalGradeDeterminator.CalculateGrade (5029 ms)
[ RUN      ] TestFinalGradeDeterminator.MultipleStudentsHandling
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] TestFinalGradeDeterminator.MultipleStudentsHandling (9 ms)
[ RUN      ] TestFinalGradeDeterminator.ClearData
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] TestFinalGradeDeterminator.ClearData (6 ms)
[----------] 9 tests from TestFinalGradeDeterminator (10331 ms total)

[----------] Global test environment tear-down
[==========] 9 tests from 1 test suite ran. (10331 ms total)
[  PASSED  ] 9 tests.
```

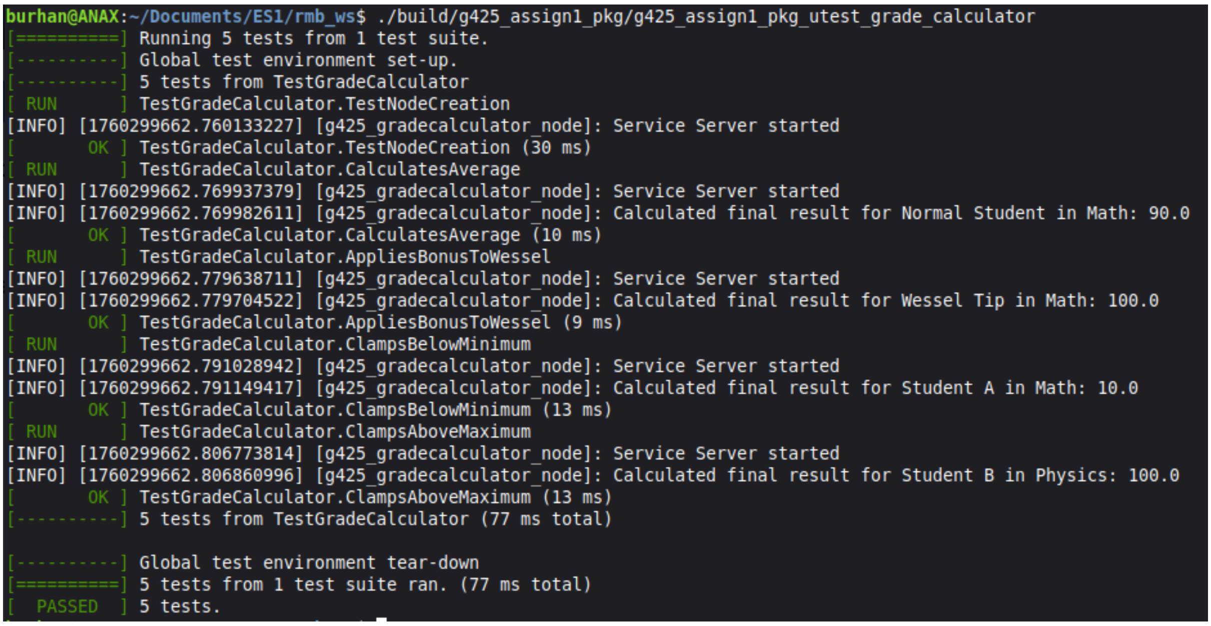
GradeCalculator

Scenarios:

- Calculate normally
- Calculate with bonus points
- Calculate above 100
- Calculate below 10

- Calculate no grades (does not do anything at the moment, as the service simply does not respond if there are no grades. If there are grades that are somehow wrongly filled, these belong to the 2 tests above)

Results:



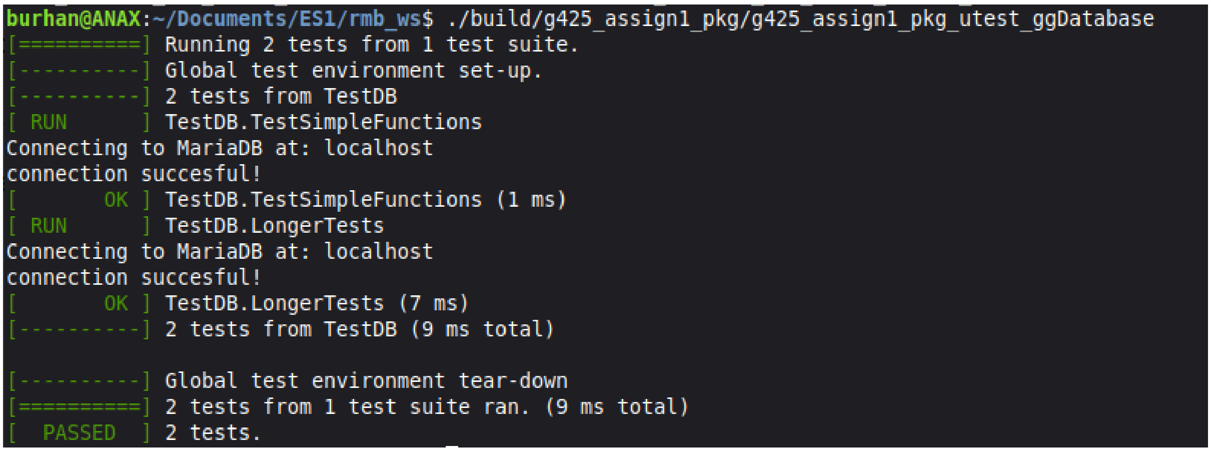
```
burhan@ANAX:~/Documents/ES1/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_grade_calculator
[==========] Running 5 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 5 tests from TestGradeCalculator
[ RUN      ] TestGradeCalculator.TestNodeCreation
[INFO] [1760299662.760133227] [g425_gradecalculator_node]: Service Server started
[       OK ] TestGradeCalculator.TestNodeCreation (30 ms)
[ RUN      ] TestGradeCalculator.CalculatesAverage
[INFO] [1760299662.769937379] [g425_gradecalculator_node]: Service Server started
[INFO] [1760299662.769982611] [g425_gradecalculator_node]: Calculated final result for Normal Student in Math: 90.0
[       OK ] TestGradeCalculator.CalculatesAverage (10 ms)
[ RUN      ] TestGradeCalculator.AppliesBonusToWessel
[INFO] [1760299662.779638711] [g425_gradecalculator_node]: Service Server started
[INFO] [1760299662.779704522] [g425_gradecalculator_node]: Calculated final result for Wessel Tip in Math: 100.0
[       OK ] TestGradeCalculator.AppliesBonusToWessel (9 ms)
[ RUN      ] TestGradeCalculator.ClampsBelowMinimum
[INFO] [1760299662.791028942] [g425_gradecalculator_node]: Service Server started
[INFO] [1760299662.791149417] [g425_gradecalculator_node]: Calculated final result for Student A in Math: 10.0
[       OK ] TestGradeCalculator.ClampsBelowMinimum (13 ms)
[ RUN      ] TestGradeCalculator.ClampsAboveMaximum
[INFO] [1760299662.806773814] [g425_gradecalculator_node]: Service Server started
[INFO] [1760299662.806860996] [g425_gradecalculator_node]: Calculated final result for Student B in Physics: 100.0
[       OK ] TestGradeCalculator.ClampsAboveMaximum (13 ms)
[----------] 5 tests from TestGradeCalculator (77 ms total)

[----------] Global test environment tear-down
[==========] 5 tests from 1 test suite ran. (77 ms total)
[  PASSED  ] 5 tests.
```

Node-Helper GGDatabase

Scenarios:

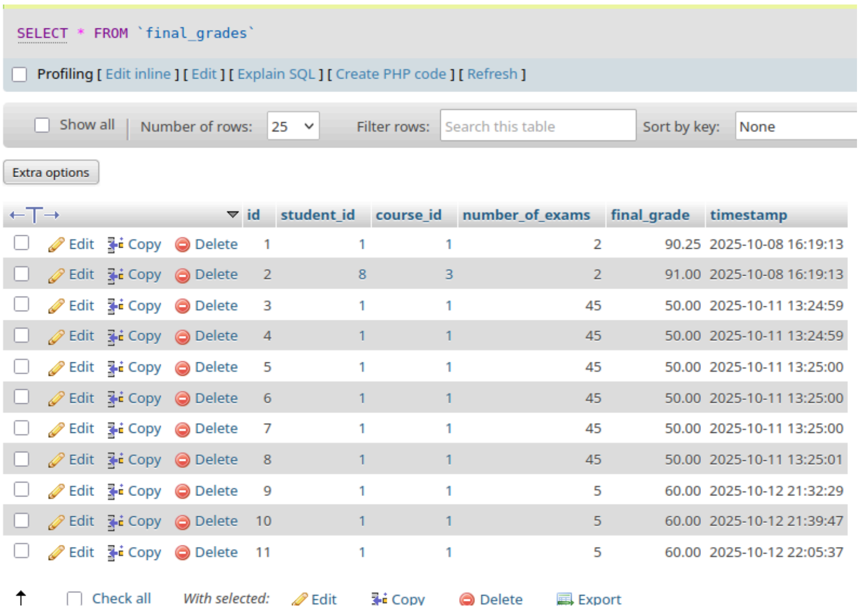
- Test all getter functions against objects that exist in the database.
- Test all setter functions by adding to the database and searching for additions where possible.
- There is no need to test with improper queries or objects in the ROS test, as they will simply be refused on the database side if the input is not allowed.

Results:



```
burhan@ANAX:~/Documents/ES1/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_utest_ggDatabase
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from TestDB
[ RUN      ] TestDB.TestSimpleFunctions
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] TestDB.TestSimpleFunctions (1 ms)
[ RUN      ] TestDB.LongerTests
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] TestDB.LongerTests (7 ms)
[----------] 2 tests from TestDB (9 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (9 ms total)
[  PASSED  ] 2 tests.
```

To show additions have been really made, result after using the setter functions repeatedly :



```
SELECT * FROM `final_grades`
```

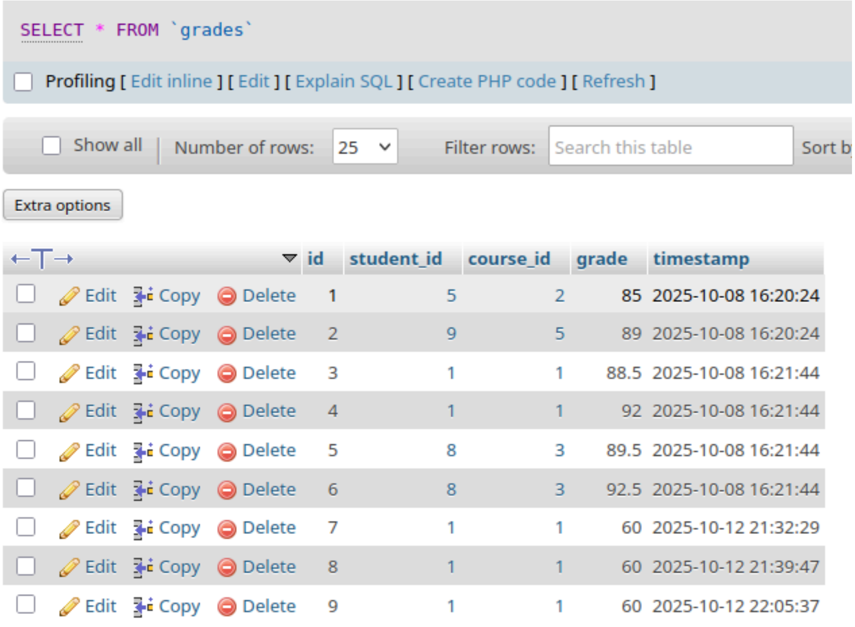
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	id	student_id	course_id	number_of_exams	final_grade	timestamp
□ Edit Copy Delete	1	1	1	2	90.25	2025-10-08 16:19:13
□ Edit Copy Delete	2	8	3	2	91.00	2025-10-08 16:19:13
□ Edit Copy Delete	3	1	1	45	50.00	2025-10-11 13:24:59
□ Edit Copy Delete	4	1	1	45	50.00	2025-10-11 13:24:59
□ Edit Copy Delete	5	1	1	45	50.00	2025-10-11 13:25:00
□ Edit Copy Delete	6	1	1	45	50.00	2025-10-11 13:25:00
□ Edit Copy Delete	7	1	1	45	50.00	2025-10-11 13:25:00
□ Edit Copy Delete	8	1	1	45	50.00	2025-10-11 13:25:01
□ Edit Copy Delete	9	1	1	5	60.00	2025-10-12 21:32:29
□ Edit Copy Delete	10	1	1	5	60.00	2025-10-12 21:39:47
□ Edit Copy Delete	11	1	1	5	60.00	2025-10-12 22:05:37

↑ □ Check all *With selected:* Edit Copy Delete Export



```
SELECT * FROM `grades`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort b

Extra options

←T→	id	student_id	course_id	grade	timestamp
□ Edit Copy Delete	1	5	2	85	2025-10-08 16:20:24
□ Edit Copy Delete	2	9	5	89	2025-10-08 16:20:24
□ Edit Copy Delete	3	1	1	88.5	2025-10-08 16:21:44
□ Edit Copy Delete	4	1	1	92	2025-10-08 16:21:44
□ Edit Copy Delete	5	8	3	89.5	2025-10-08 16:21:44
□ Edit Copy Delete	6	8	3	92.5	2025-10-08 16:21:44
□ Edit Copy Delete	7	1	1	60	2025-10-12 21:32:29
□ Edit Copy Delete	8	1	1	60	2025-10-12 21:39:47
□ Edit Copy Delete	9	1	1	60	2025-10-12 22:05:37

System Test (includes integration test)

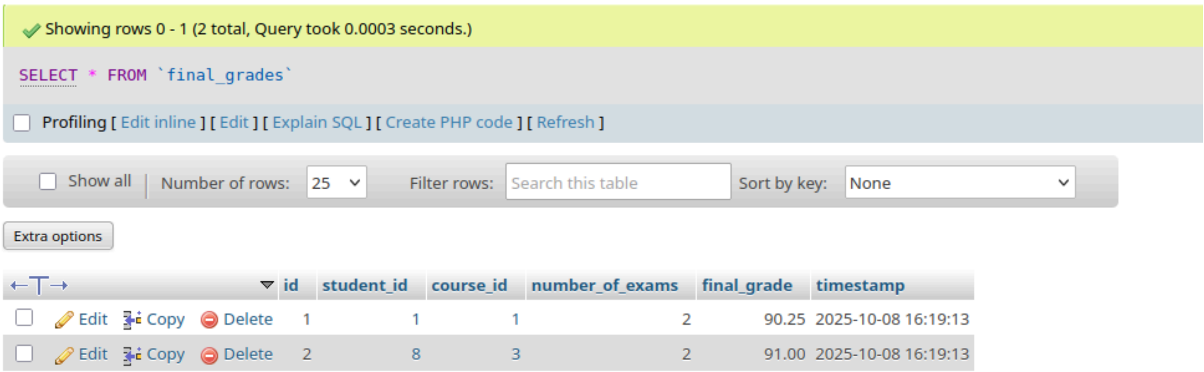
Does the entire system work together?

Full program test

This can be achieved by simply running the system and checking the output, a little helper test node has been created but is not mandatory.

Results:

Before running the system:



✓ Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

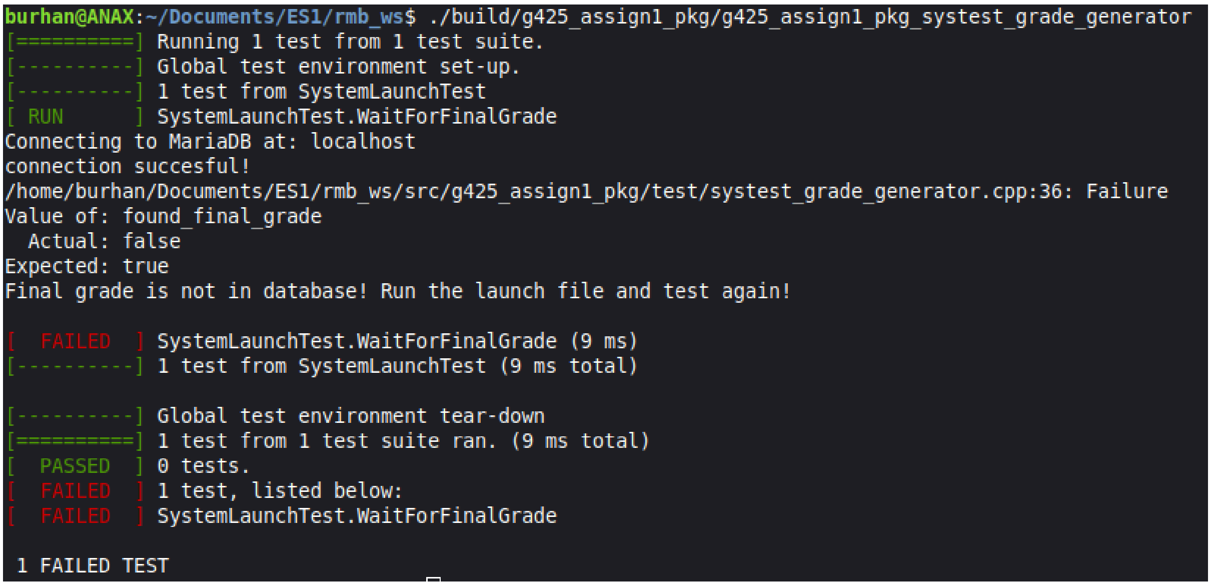
```
SELECT * FROM `final_grades`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

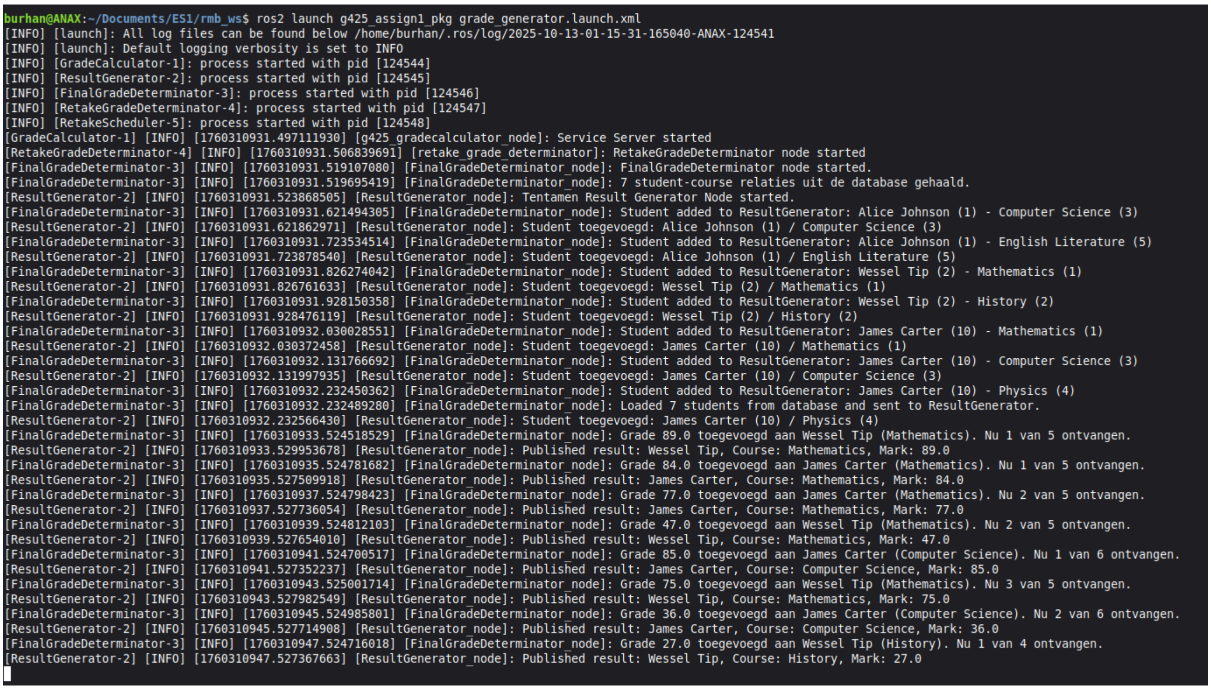
Extra options

←T→	id	student_id	course_id	number_of_exams	final_grade	timestamp
□ Edit Copy Delete	1	1	1	2	90.25	2025-10-08 16:19:13
□ Edit Copy Delete	2	8	3	2	91.00	2025-10-08 16:19:13

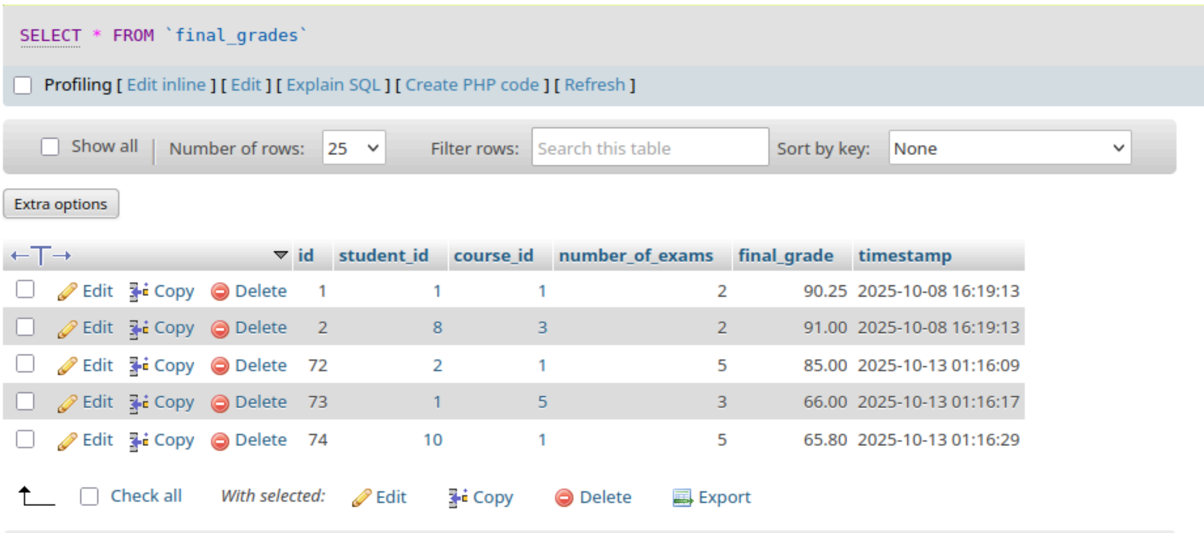


```
burhan@ANAX:~/Documents/ES1/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_systest_grade_generator
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from SystemLaunchTest
[ RUN      ] SystemLaunchTest.WaitForFinalGrade
Connecting to MariaDB at: localhost
connection succesful!
/home/burhan/Documents/ES1/rmb_ws/src/g425_assign1_pkg/test/systest_grade_generator.cpp:36: Failure
Value of: found_final_grade
  Actual: false
Expected: true
Final grade is not in database! Run the launch file and test again!

[  FAILED  ] SystemLaunchTest.WaitForFinalGrade (9 ms)
[----------] 1 test from SystemLaunchTest (9 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (9 ms total)
[  PASSED  ] 0 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] SystemLaunchTest.WaitForFinalGrade

 1 FAILED TEST
```

We then run the launch file and watch the results come in on our mysql GUI of choice:



```
burhan@ANAX:~/Documents/ES1/rmb_ws$ ros2 launch g425_assign1_pkg grade_generator.launch.xml
[INFO] [launch]: All log files can be found below /home/burhan/.ros/log/2025-10-13-01-15-31-165040-ANAX-124541
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [GradeCalculator-1]: process started with pid [124544]
[INFO] [ResultGenerator-2]: process started with pid [124545]
[INFO] [FinalGradeDeterminator-3]: process started with pid [124546]
[INFO] [RetakeGradeDeterminator-4]: process started with pid [124547]
[INFO] [RetakeScheduler-5]: process started with pid [124548]
[GradeCalculator-1] [INFO] [1760310931.497111930] [g425_gradecalculator_node]: Service Server started
[RetakeGradeDeterminator-4] [INFO] [1760310931.506839691] [retake_grade_determinator]: RetakeGradeDeterminator node started
[FinalGradeDeterminator-3] [INFO] [1760310931.519107080] [FinalGradeDeterminator_node]: FinalGradeDeterminator node started.
[FinalGradeDeterminator-3] [INFO] [1760310931.519695419] [FinalGradeDeterminator_node]: 7 student-course relaties uit de database gehaald.
[ResultGenerator-2] [INFO] [1760310931.523868505] [ResultGenerator_node]: Tentamen Result Generator Node started.
[FinalGradeDeterminator-3] [INFO] [1760310931.621494305] [FinalGradeDeterminator_node]: Student added to ResultGenerator: Alice Johnson (1) - Computer Science (3)
[ResultGenerator-2] [INFO] [1760310931.621862971] [ResultGenerator_node]: Student toegevoegd: Alice Johnson (1) / Computer Science (3)
[FinalGradeDeterminator-3] [INFO] [1760310931.723534514] [FinalGradeDeterminator_node]: Student added to ResultGenerator: Alice Johnson (1) - English Literature (5)
[ResultGenerator-2] [INFO] [1760310931.723878540] [ResultGenerator_node]: Student toegevoegd: Alice Johnson (1) / English Literature (5)
[FinalGradeDeterminator-3] [INFO] [1760310931.826274042] [FinalGradeDeterminator_node]: Student added to ResultGenerator: Wessel Tip (2) - Mathematics (1)
[ResultGenerator-2] [INFO] [1760310931.826761633] [ResultGenerator_node]: Student toegevoegd: Wessel Tip (2) / Mathematics (1)
[FinalGradeDeterminator-3] [INFO] [1760310931.928150358] [FinalGradeDeterminator_node]: Student added to ResultGenerator: Wessel Tip (2) - History (2)
[ResultGenerator-2] [INFO] [1760310931.928476119] [ResultGenerator_node]: Student toegevoegd: Wessel Tip (2) / History (2)
[FinalGradeDeterminator-3] [INFO] [1760310932.030028551] [FinalGradeDeterminator_node]: Student added to ResultGenerator: James Carter (10) - Mathematics (1)
[ResultGenerator-2] [INFO] [1760310932.030372458] [ResultGenerator_node]: Student toegevoegd: James Carter (10) / Mathematics (1)
[FinalGradeDeterminator-3] [INFO] [1760310932.131766692] [FinalGradeDeterminator_node]: Student added to ResultGenerator: James Carter (10) - Computer Science (3)
[ResultGenerator-2] [INFO] [1760310932.131997935] [ResultGenerator_node]: Student toegevoegd: James Carter (10) / Computer Science (3)
[FinalGradeDeterminator-3] [INFO] [1760310932.232450362] [FinalGradeDeterminator_node]: Student added to ResultGenerator: James Carter (10) - Physics (4)
[FinalGradeDeterminator-3] [INFO] [1760310932.232489280] [FinalGradeDeterminator_node]: Loaded 7 students from database and sent to ResultGenerator.
[ResultGenerator-2] [INFO] [1760310932.232566430] [ResultGenerator_node]: Student toegevoegd: James Carter (10) / Physics (4)
[FinalGradeDeterminator-3] [INFO] [1760310933.524518529] [FinalGradeDeterminator_node]: Grade 89.0 toegevoegd aan Wessel Tip (Mathematics). Nu 1 van 5 ontvangen.
[ResultGenerator-2] [INFO] [1760310933.529953678] [ResultGenerator_node]: Published result: Wessel Tip, Course: Mathematics, Mark: 89.0
[FinalGradeDeterminator-3] [INFO] [1760310935.524781682] [FinalGradeDeterminator_node]: Grade 84.0 toegevoegd aan James Carter (Mathematics). Nu 1 van 5 ontvangen.
[ResultGenerator-2] [INFO] [1760310935.527509918] [ResultGenerator_node]: Published result: James Carter, Course: Mathematics, Mark: 84.0
[FinalGradeDeterminator-3] [INFO] [1760310937.524798423] [FinalGradeDeterminator_node]: Grade 77.0 toegevoegd aan James Carter (Mathematics). Nu 2 van 5 ontvangen.
[ResultGenerator-2] [INFO] [1760310937.527736054] [ResultGenerator_node]: Published result: James Carter, Course: Mathematics, Mark: 77.0
[FinalGradeDeterminator-3] [INFO] [1760310939.524812103] [FinalGradeDeterminator_node]: Grade 47.0 toegevoegd aan Wessel Tip (Mathematics). Nu 2 van 5 ontvangen.
[ResultGenerator-2] [INFO] [1760310939.527654010] [ResultGenerator_node]: Published result: Wessel Tip, Course: Mathematics, Mark: 47.0
[FinalGradeDeterminator-3] [INFO] [1760310941.524700517] [FinalGradeDeterminator_node]: Grade 85.0 toegevoegd aan James Carter (Computer Science). Nu 1 van 6 ontvangen.
[ResultGenerator-2] [INFO] [1760310941.527352237] [ResultGenerator_node]: Published result: James Carter, Course: Computer Science, Mark: 85.0
[FinalGradeDeterminator-3] [INFO] [1760310943.525001714] [FinalGradeDeterminator_node]: Grade 75.0 toegevoegd aan Wessel Tip (Mathematics). Nu 3 van 5 ontvangen.
[ResultGenerator-2] [INFO] [1760310943.527982549] [ResultGenerator_node]: Published result: Wessel Tip, Course: Mathematics, Mark: 75.0
[FinalGradeDeterminator-3] [INFO] [1760310945.524985801] [FinalGradeDeterminator_node]: Grade 36.0 toegevoegd aan James Carter (Computer Science). Nu 2 van 6 ontvangen.
[ResultGenerator-2] [INFO] [1760310945.527714908] [ResultGenerator_node]: Published result: James Carter, Course: Computer Science, Mark: 36.0
[FinalGradeDeterminator-3] [INFO] [1760310947.524716018] [FinalGradeDeterminator_node]: Grade 27.0 toegevoegd aan Wessel Tip (History). Nu 1 van 4 ontvangen.
[ResultGenerator-2] [INFO] [1760310947.527367663] [ResultGenerator_node]: Published result: Wessel Tip, Course: History, Mark: 27.0
```



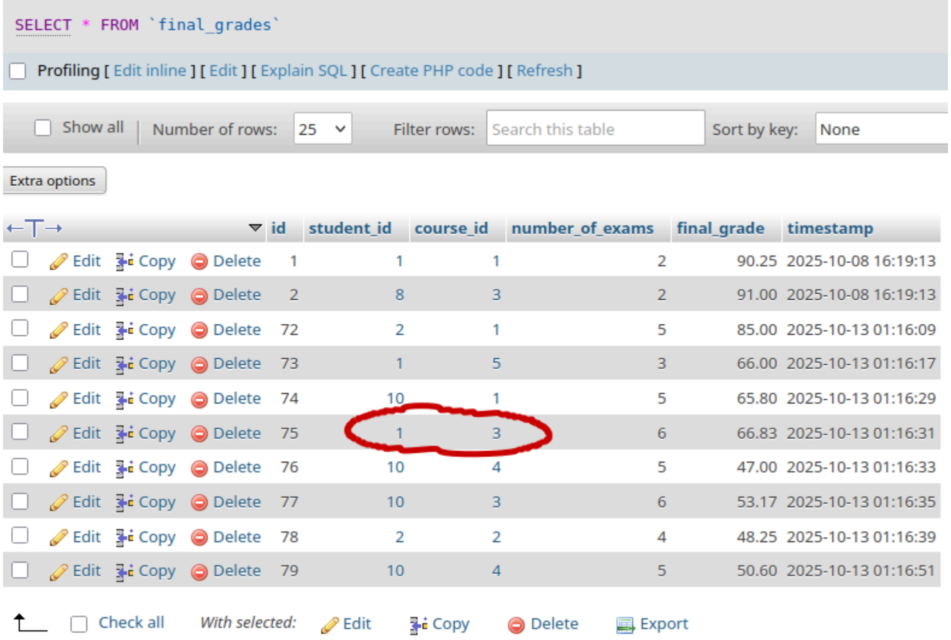
```
SELECT * FROM `final_grades`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

←T→	id	student_id	course_id	number_of_exams	final_grade	timestamp
□ Edit Copy Delete	1	1	1	2	90.25	2025-10-08 16:19:13
□ Edit Copy Delete	2	8	3	2	91.00	2025-10-08 16:19:13
□ Edit Copy Delete	72	2	1	5	85.00	2025-10-13 01:16:09
□ Edit Copy Delete	73	1	5	3	66.00	2025-10-13 01:16:17
□ Edit Copy Delete	74	10	1	5	65.80	2025-10-13 01:16:29

When whatever student and course we are searching for appears here we can stop the system. Of course if it is hard to track we can simply keep the system running until our test returns without errors.



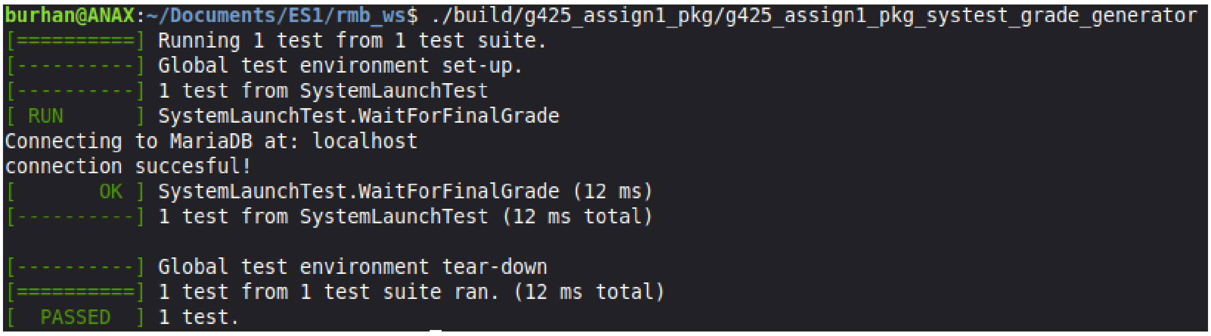
```
SELECT * FROM `final_grades`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

				id	student_id	course_id	number_of_exams	final_grade	timestamp
□	Edit	Copy	Delete	1	1	1	2	90.25	2025-10-08 16:19:13
□	Edit	Copy	Delete	2	8	3	2	91.00	2025-10-08 16:19:13
□	Edit	Copy	Delete	72	2	1	5	85.00	2025-10-13 01:16:09
□	Edit	Copy	Delete	73	1	5	3	66.00	2025-10-13 01:16:17
□	Edit	Copy	Delete	74	10	1	5	65.80	2025-10-13 01:16:29
□	Edit	Copy	Delete	75	1	3	6	66.83	2025-10-13 01:16:31
□	Edit	Copy	Delete	76	10	4	5	47.00	2025-10-13 01:16:33
□	Edit	Copy	Delete	77	10	3	6	53.17	2025-10-13 01:16:35
□	Edit	Copy	Delete	78	2	2	4	48.25	2025-10-13 01:16:39
□	Edit	Copy	Delete	79	10	4	5	50.60	2025-10-13 01:16:51

And then we perform the test again.



```
burhan@ANAX:~/Documents/ES1/rmb_ws$ ./build/g425_assign1_pkg/g425_assign1_pkg_systest_grade_generator
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from SystemLaunchTest
[ RUN      ] SystemLaunchTest.WaitForFinalGrade
Connecting to MariaDB at: localhost
connection succesful!
[       OK ] SystemLaunchTest.WaitForFinalGrade (12 ms)
[----------] 1 test from SystemLaunchTest (12 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (12 ms total)
[  PASSED  ] 1 test.
```

Acceptance Test

Will the system work for a typical user?

A typical user in this case would mean the school faculty adds the students and the courses to their database, and a teacher adds grades in the administration tool.
For this use case, the database works as intended.
Once a grade is sent in, the administration tool should simply act similar to the Result Generator. Only instead of random results, we of course add an exact grade.

The workflow needs to replace the Result Generator with a simple Subscriber node that adds grades to the database and calls the Grade Determinators.

```
...
add_student_sub_ = this->create_subscription<Student>(
       "add_students", 10,
       std::bind(&ResultGenerator::add_student, this, _1));

void publish_random_result()
   {
       ...
       db_.addGrade(grade);
   }
   ...
   // Publisher (asks ResultGenerator for new random numbers)
   publisher_ = this->create_publisher<Student>("add_students", 10);
```

After that, the system will work in production.