
CoursesManagementApp

Sprint Report

SOTIRIOS PANAGIOTOU 4456, cs04456@uoi.gr
MYRITZIS EFSTRATIOS 4444, cs04444@uoi.gr
DIMITRIOS GIANNAKOPOULOS 4336, cs04336@uoi.gr

*Για να τρέξουν τα statistics πρέπει να γίνει import του commons-math3.jar library

*Για να passαρουν όλα τα τεστς αρχικοποιουμε τη βάση

Github: <https://github.com/sotblad/SE-Project>

Demo video: <https://github.com/sotblad/SE-Project/blob/main/zarr.mp4>

VERSIONS HISTORY

Date	Version	Description	Author
15/5/22	1.0	User stories implemented. Unit tests all pass without any errors. Everything engineered with maintainability and usability in mind.	SOTIRIOS PANAGIOTOU MYRITZIS EFSTATIOS DIMITRIS GIANNAKOPOULOS

1 Introduction

This document provides information concerning the **3** sprints of the project.

The objective of this project is to develop a Web application that allows an instructor to manage the grading of the courses that he teaches.

1.1 Document Structure

The rest of this document is structured as follows. Section 2 describes out Scrum team and specifies the Sprint's backlog. Section 3 specifies the main design concepts for this release of the project.

2 Scrum team and Sprint Backlog

2.1 Scrum team

Product Owner	Zarras
Scrum Master	-
Development Team	SOTIRIOS PANAGIOTOU MYRITZIS EFSTATIOS DIMITRIS GIANNAKOPOULOS

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
S1	3/5/22	4/5/22	0	US1,US1.1,US2,US3,US4,US5
S2	4/5/22	5/5/22	0	US6,US7,US8,US9,US10
S3	5/5/22	6/5/22	0	US11,US12

3 Use Cases

3.1 <Use Case 1>

Use case ID	1
Actors	Instructor
Pre conditions	For the correct execution of US1, a database has to be created and inside it there must be stored the user names and passwords of the users.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the user loads the /login page.2. There is a form where the user is able to fill in his info(username, password).
Post conditions	After the correct execution of this use case, the user has access to the list of courses being taught by him.

3.1.1 <Use Case 1.1>

Use case ID	1.1
Actors	Instructor
Pre conditions	For the correct execution of US1.1, US1 has to be implemented. And in the /login page there is a button for the user to be redirected in the /register page.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the user loads the /register page.2. There is a form where the user must be able to fill in his info.3. The info of the new user is added to the user database.
Post conditions	After the correct execution of this use case, the user is redirected back to the /login page .

3.2 <Use Case 2>

Use case ID	2
Actors	Instructor
Pre conditions	For the correct execution of US2, US1 has to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after inserting his credentials in the login form and being redirected in the /myCourses page.2. On each course there are buttons to:<ol style="list-style-type: none">2.1 View the course,2.2 Edit the course info,2.3 Delete the course.
Post conditions	After the correct execution of this use case, the user is able to manage the courses being taught by him.

3.3 <Use Case 3>

Use case ID	3
Actors	Instructor
Pre conditions	For the correct execution of US3, US1 and US2 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the blue “Add Course” button in the /myCourses page.2. Then the user is redirected to the /addCourse page where he is asked to fill in the following information about the course:<ol style="list-style-type: none">2.1 Course name,2.2 Syllabus,2.3 Year,2.4 Semester.
Post conditions	After the correct execution of this use case, the new course is added to the instructor’s courses.

3.4 <Use Case 4>

Use case ID	4
Actors	Instructor
Pre conditions	For the correct execution of US4, US1, US2 and US3 have to be implemented.
Main flow of events	1. The use case starts after clicking the red “Trash Can” button in the /myCourses page.
Post conditions	After the correct execution of this use case, the selected course is removed from the courses being taught by the instructor.

3.5 <Use Case 5>

Use case ID	5
Actors	Instructor
Pre conditions	For the correct execution of US5, US1, US2 and US3 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the green button in the /myCourses page.2. Then the user is redirected in the /editCourse page where he is asked to edit the following information about the course:<ol style="list-style-type: none">2.1 Course name,2.2 Syllabus,2.3 Year,2.4 Semester.
Post conditions	After the correct execution of this use case, the selected course’s information is updated.

3.6 <Use Case 6>

Use case ID	6
Actors	Instructor
Pre conditions	For the correct execution of US6, US1, US2 and US3 have to be implemented.
Main flow of events	1. The use case starts after clicking the blue “Arrow” button in the /myCourses page.
Post conditions	After the correct execution of this use case, the instructor is redirected to the /viewCourse page of the selected course where there is a list of students enrolled in the specific course.

3.7 <Use Case 7>

Use case ID	7
Actors	Instructor
Pre conditions	For the correct execution of US7, US1, US2, US3 and US6 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the blue “Add student” button in the /viewCourse page.2. Then the user is redirected in the /addStudent page where he is asked to fill in the information of the new student:<ol style="list-style-type: none">2.1 Student ID,2.2 Student Name,2.3 Year of registration,2.4 Semester.
Post conditions	After the correct execution of this use case, the instructor is redirected back to the selected course where he should be able to see the new student.

3.8 <Use Case 8>

Use case ID	8
Actors	Instructor
Pre conditions	For the correct execution of US8, US1, US2, US3, US6 and US7 have to be implemented.
Main flow of events	1. The use case starts after clicking the red “Delete” button in the /viewCourse page next to a student.
Post conditions	After the correct execution of this use case, the instructor is redirected back to the selected course where the deleted student should be missing.

3.9 <Use Case 9>

Use case ID	9
Actors	Instructor
Pre conditions	For the correct execution of US9, US1, US2, US3, US6 and US7 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the blue “Edit Student” button in the /viewCourse page.2. Then the instructor is redirected in the /editStudent page where he is asked to edit the following information about the student:<ol style="list-style-type: none">2.1 Student ID,2.2 Student Name,2.3 Year of registration,3. Finally the user has the option to either Soft update or Hard update.<ol style="list-style-type: none">3.1. When Soft updating, the student’s new information is only updated in the specific course page.3.2. When Hard updating, the student’s new information is updated in every course the student participates.
Post conditions	After the correct execution of this use case, the instructor is redirected back to the selected course where the student’s information is updated with the new values.

3.10 <Use Case 10>

Use case ID	10
Actors	Instructor
Pre conditions	For the correct execution of US10, US1, US2, US3, US6 and US7 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the green “Edit Grades” button in the /viewCourse page.2. Then the user is redirected to the /editGrades page where he is asked to edit the following information about the student:<ol style="list-style-type: none">2.1 Exam grade,2.2 Project grade
Post conditions	After the correct execution of this use case, the instructor is redirected back to the selected course and the student’s grades are updated with the new values.

3.11 <Use Case 11>

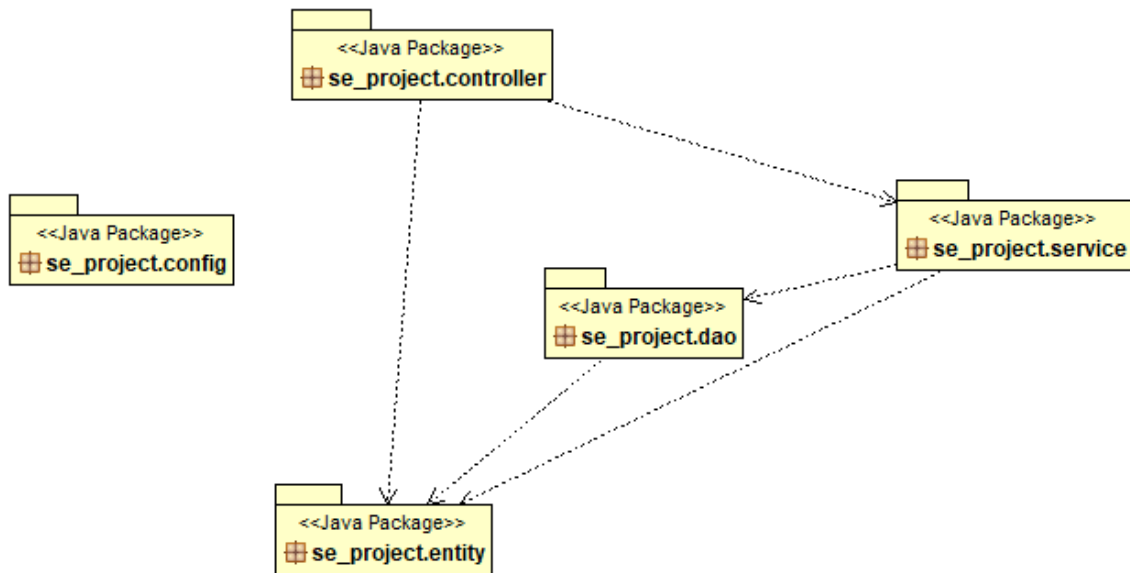
Use case ID	11
Actors	Instructor
Pre conditions	For the correct execution of US11, US1, US2, US3, US6, US7 and US10 have to be implemented. The user must have clicked the green “Edit Weights” button and have entered the exam-project weights. Also the instructor should have pressed the green “Edit Grades” button to enter the corresponding grades.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the blue “Calculate Grades” button in the /viewCourse page.
Post conditions	After the correct execution of this use case, the user is on the /viewCourse page where the student’s grade is updated with the generated values.

3.12 <Use Case 12>

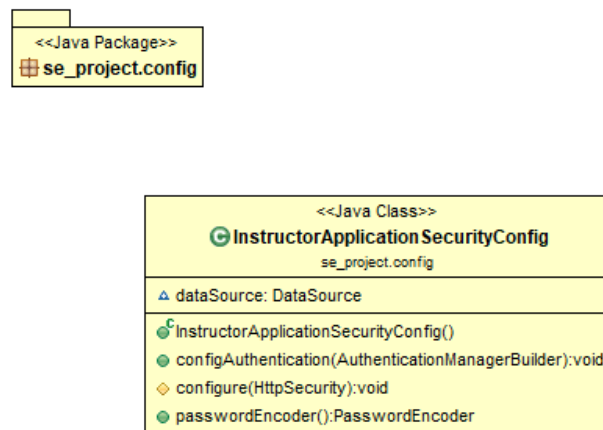
Use case ID	12
Actors	Instructor
Pre conditions	For the correct execution of US12, US1, US2, US3, US6, US7, US10 and US11 have to be implemented.
Main flow of events	<ol style="list-style-type: none">1. The use case starts after clicking the red “View Stats” button in the /viewCourse page.2. The instructor is redirected to the /viewStats page where he can see the generated descriptive statistics about the grades of the students enrolled in the course. The statistics include:<ol style="list-style-type: none">2.1 Min.Grade2.2 Max. Grade2.3 Mean Grade2.4 Std. Deviation2.5 Variance2.6 Skewness2.7 Kurtosis2.8 Median2.9 Percentiles
Post conditions	After the execution of this use case, the statistics get calculated and displayed on the /viewStats page of the selected course.

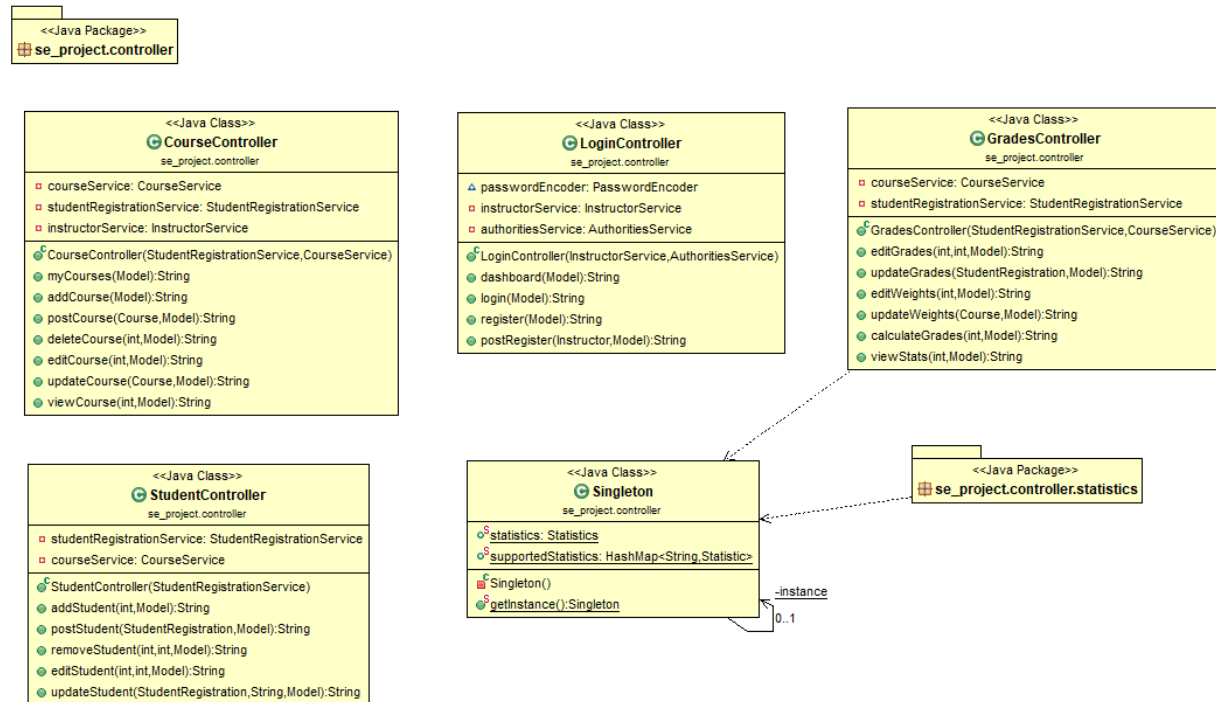
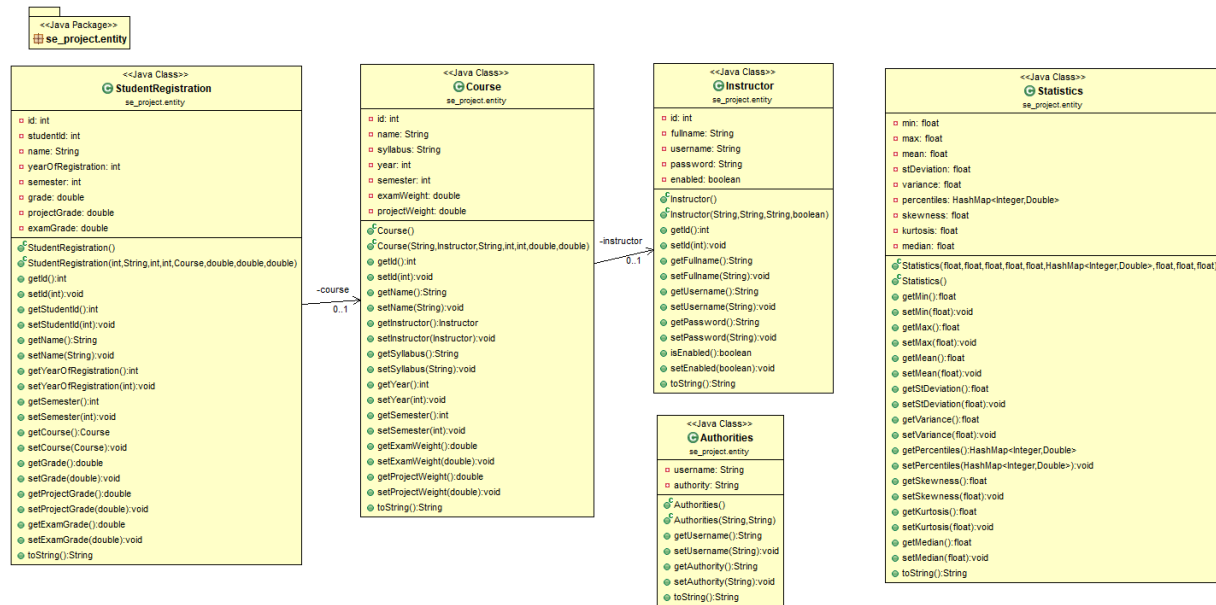
4 Design

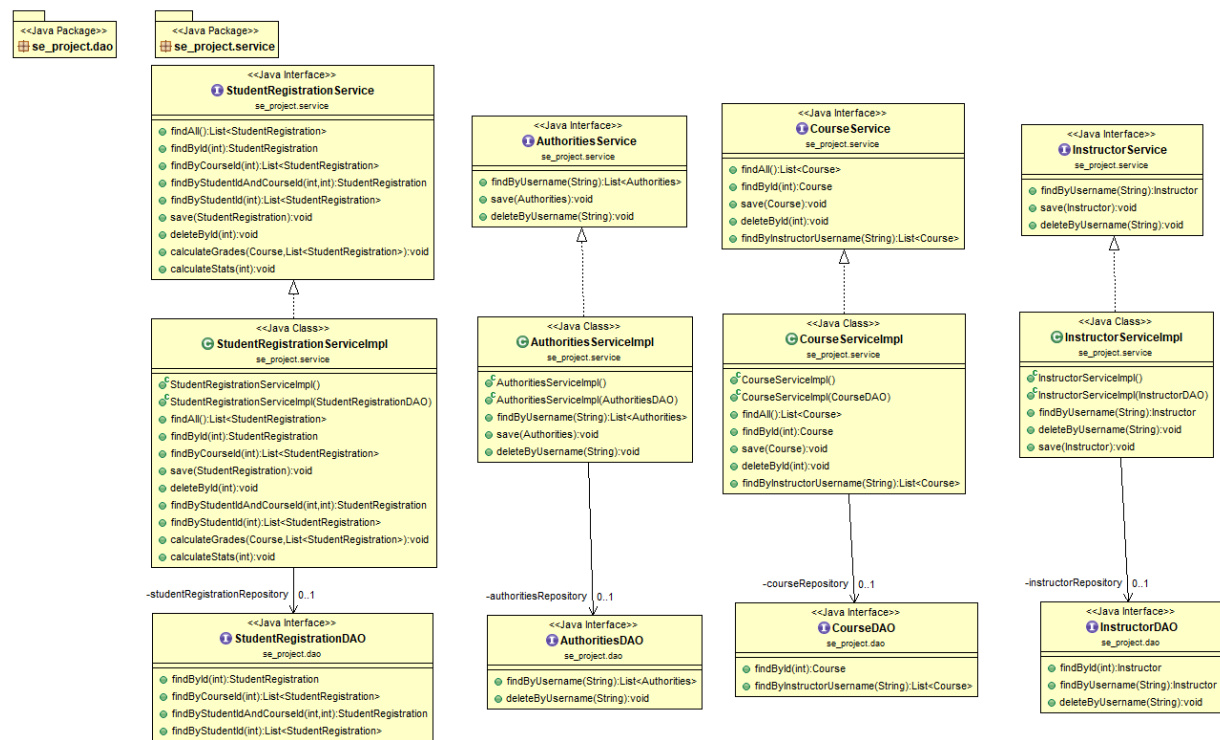
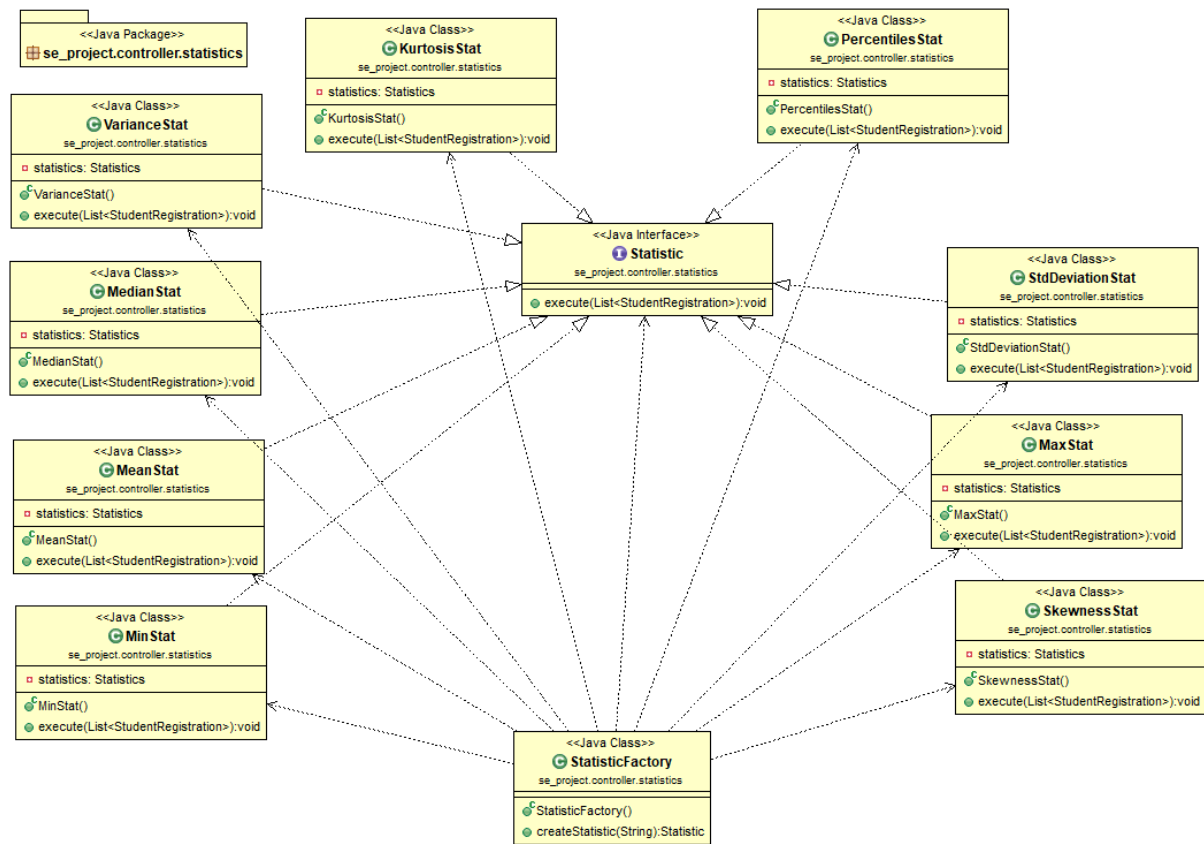
4.1 Architecture

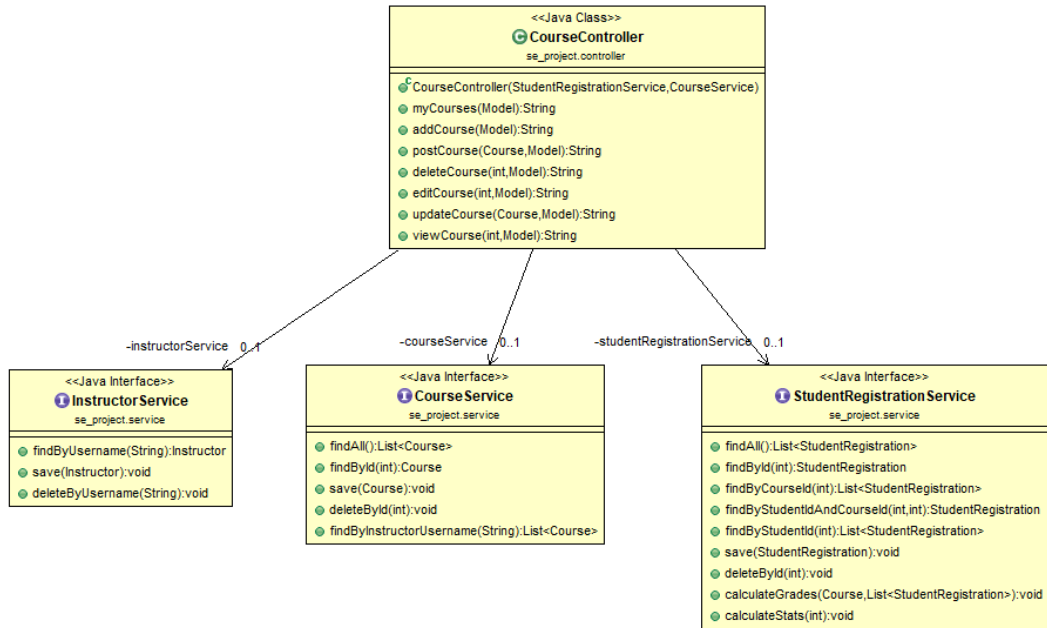


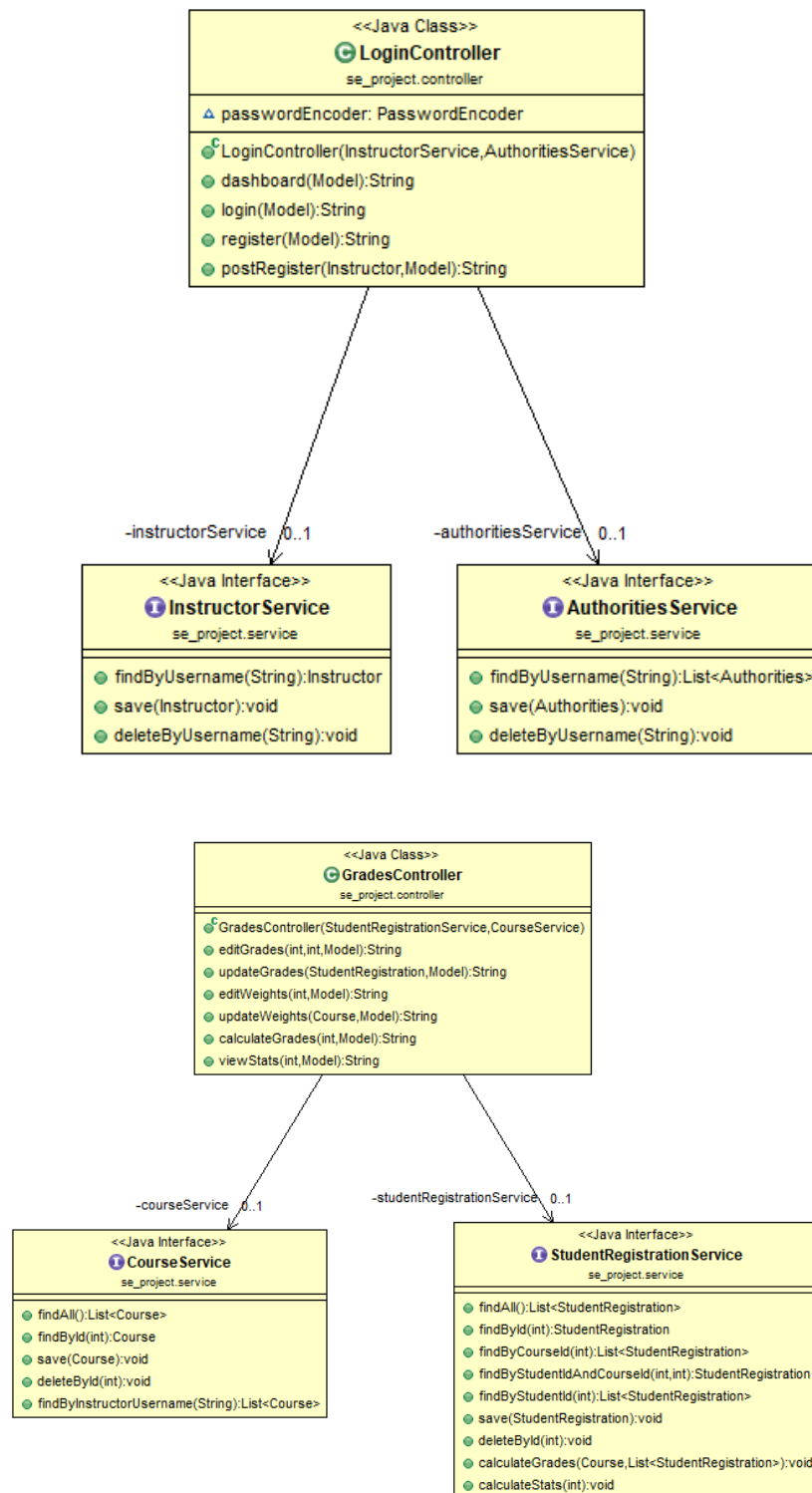
4.2 Design











Class Name: Authorities	
Responsibilities: <ul style="list-style-type: none"> This class creates authority objects and holds their properties 	Collaborations: <ul style="list-style-type: none"> This class is used by the Authorities DAO to store authorities and grab them from the database. This is also used by the security for the login.

Class Name: Course	
Responsibilities: <ul style="list-style-type: none"> This class creates course objects and holds their properties 	Collaborations: <ul style="list-style-type: none"> This class is used by the Course DAO to store courses and grab them from the database. This is also used by the CourseController to implement the courses related user stories.

Class Name: Instructor	
Responsibilities: <ul style="list-style-type: none"> This class creates instructor objects and holds their properties 	Collaborations: <ul style="list-style-type: none"> This class is used by the Instructor DAO to store instructors and grab them from the database. This is also used by the LoginController and the security for the login & registration.

Class Name: StudentRegistration	
Responsibilities: <ul style="list-style-type: none"> This class creates student registration objects and holds their properties 	Collaborations: <ul style="list-style-type: none"> This class is used by the StudentRegistration DAO to store student registrations and grab them from the database. This is also used by the StudentController to implement the student related user stories.

Class Name: Statistics	
Responsibilities: <ul style="list-style-type: none"> ▪ This class creates Statistics objects and holds their properties 	Collaborations: <ul style="list-style-type: none"> ▪ This class is used by the GradesController and the StudentRegistration services to calculate and view the statistics.

Class Name: Singleton	
Responsibilities: <ul style="list-style-type: none"> ▪ Holds shared fields used by multiple other classes (statistics) 	Collaborations: <ul style="list-style-type: none"> ▪ Collaborates with all the statistics classes. ▪ It also collaborates with the StatisticFactory class, which is getting created on the Singleton class.

Class Name: StatisticFactory	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates the supported statistics. 	Collaborations: <ul style="list-style-type: none"> ▪ Collaborates with all the statistics classes. ▪ It gets executed from the singleton class to instantiate the statistics.