

---

# CoursesManagementApp

## Sprint Report

---

Team name: GoTI

Ζάγκας Δημοσθένης 4359

Ψαλίδας Ζήσης 3369

Βάρελης Παναγιώτης 3388

---

## VERSIONS HISTORY

---

Date	Version	Description	Author
2/05	1.1	First Sprint	GoTI
10/05	1.2	Second Sprint	GoTI
16/05	1.3	Third Sprint	GoTI
18/05	1.4	Extra Details	GoTI

## 1 Introduction

---

This document provides information concerning the <X> sprint of the project.

### 1.1 Purpose

### 1.2 Document Structure

---

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

## 2 Scrum team and Sprint Backlog

---

<For the user stories included in this release specify below corresponding tests using a typical tabular form.>

### 2.1 Scrum team

---

<b>Product Owner</b>	Zarras
<b>Scrum Master</b>	....
<b>Development Team</b>	GoTI (the Gank of Three Idiots)

### 2.2 Sprints

---

<List below the sprints that you performed and the user stories that have been realized in each Sprint>

<b>Sprint No</b>	<b>Begin Date</b>	<b>End Date</b>	<b>Number of weeks</b>	<b>User stories</b>
1	25/04	2/05	1	US1,US2
2	03/5	10/05	1	US3,US4,US5

3	11/05	16/05	1	US6,US7,US8,US9,US10
4	16/05	18/05	-	Extra Details/Fixes

### 3 Use Cases

---

<Specify the concrete Use Cases that describe the interaction of the user with the applications, as derived from the abstract user stories. Give a **UML Use Case diagram** and the **detailed use case descriptions**.>

#### 3.1 <Use Case 1>

---

<b>Use case ID</b>	US1
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Knowing the username and the password.
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case starts when the user types the link to the application</li> <li>2. A log in page appears</li> <li>3. The user inputs the credentials <ol style="list-style-type: none"> <li>3.1. Correct credentials: The user is sent to the page with the list of the courses</li> <li>3.2. Incorrect credentials: The user is prompted with a message "incorrect credentials"</li> </ol> </li> </ol>
<b>Alternative flow 1</b>	[Alternative flow of events IF ANY that takes place in case of an exception]
<b>Alternative flow 2</b>	.....
<b>Post conditions</b>	

#### 3.2 <Use Case 2>

---

<b>Use case ID</b>	US2
<b>Actors</b>	Instructor

<b>Pre conditions</b>	Having logged in A course existing (for updating information for the course)
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case starts when the user logs in successfully</li> <li>2. The list of the courses appears</li> <li>3. Under the “Actions” tab, the user has the option to “Update” the courses information</li> <li>4. The update button sends the user to a new page, with the fields of the course (name, description etc), filled in.</li> <li>5. The user can make changes and the save them.</li> </ol>
<b>Alternative flow 1</b>	For the user to change the students grades, the user needs to open the student list and access each student separately. Will be described in a different US
<b>Alternative flow 2</b>	.....
<b>Post conditions</b>	

### 3.3 <Use Case 3>

---

<b>Use case ID</b>	US3
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Having logged in
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case starts when the user clicks the “add course” button</li> <li>2. A new page loads, showing the empty fields of a course (name, year, etc.)</li> <li>3. The user fills in the fields</li> <li>4. Presses the save button</li> <li>5. The page with the courses list loads, showing the new course created</li> </ol>
<b>Alternative flow 1</b>	The user has the option to return to the course list at any time by pressing the “return” button
<b>Alternative flow 2</b>	The user has the option to not fill in a field, in order to update it later on
<b>Post</b>	A new course has been added

<b>conditions</b>	
-------------------	--

### 3.4 <Use Case 4>

---

<b>Use case ID</b>	US4
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Having logged in A course existing in the list
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case starts when the user clicks the “delete” button in the action tab on a course</li> <li>2. A message pops up, asking for confirmation</li> </ol>
<b>Alternative flow 1</b>	
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The course gets deleted

### 3.5 <Use Case 5>

---

<b>Use case ID</b>	US5
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Having logged in A course existing in the list
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The use case starts when the user clicks the “update” button at the action tab</li> <li>2. The update button sends the user to a new page, with the fields of the course (name, description etc), filled in.</li> <li>3. The user can make changes and the save them.</li> </ol>

<b>Alternative flow 1</b>	The user has the option to return to the list without making any changes by clicking the “return” button
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The course info gets updated

### 3.6 <Use Case 6>

---

<b>Use case ID</b>	US6
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Having logged in A course existing in the list
<b>Main flow of events</b>	1. The use case starts when the user clicks the “show students” button at the action tab 2. A new page loads, with the students of that course
<b>Alternative flow 1</b>	
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The page with the list of the students loads

### 3.7 <Use Case 7>

---

<b>Use case ID</b>	US7
<b>Actors</b>	Instructor
<b>Pre conditions</b>	Having opened the page with the students list
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The user story begins when the user clicks the “add student” button</li><li>2. A new page loads, with the fields (name, year, etc) of the student empty</li><li>3. The user fills in the fields and clicks save</li><li>4. A new student has been created in that course</li></ol>
<b>Alternative flow 1</b>	The user has the option to return to the list without making any changes by clicking the “return” button
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The course info gets populated with an extra student

### 3.8 <Use Case 8>

---

<b>Use case ID</b>	US8
<b>Actors</b>	Instructor
<b>Pre conditions</b>	A student existing in the course
<b>Main flow of events</b>	<ol style="list-style-type: none"><li>1. The user story starts when the user clicks the “delete” button at a student</li><li>2. A confirmation message appears</li><li>3. The user clicks yes and the student gets deleted</li></ol>
<b>Alternative flow 1</b>	The user has the option to click no at the confirmation message
<b>Alternative flow 2</b>	



<b>Post conditions</b>	The student gets deleted
------------------------	--------------------------

### 3.9 <Use Case 9>

---

<b>Use case ID</b>	US5
<b>Actors</b>	Instructor
<b>Pre conditions</b>	A student existing in the course
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The user story starts when the user clicks the “update” button at a student</li> <li>2. A new page loads with the student fields already filled in</li> <li>3. The user can change each field</li> <li>4. The user clicks save</li> <li>5. The page with the list of the students loads</li> </ol>
<b>Alternative flow 1</b>	The user has the option to return to the list without making any changes by clicking the “return” button
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The student info gets updated

### 3.10 <Use Case 5>

---

<b>Use case ID</b>	US10
<b>Actors</b>	Instructor
<b>Pre conditions</b>	A student existing in the course
<b>Main flow of events</b>	<ol style="list-style-type: none"> <li>1. The user story begins when the user clicks the “update” button in the action tab of the student</li> </ol>

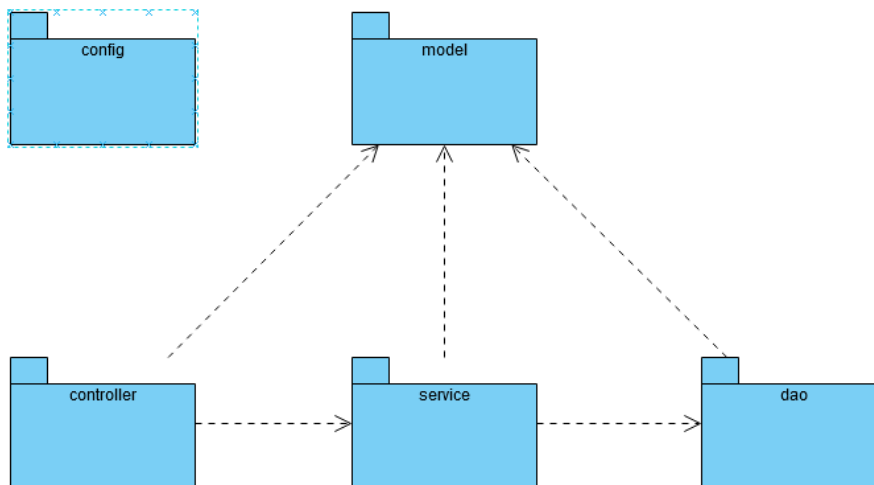
	<ol style="list-style-type: none"> <li>2. The page with the student info loads</li> <li>3. The user can change the grades of the student</li> <li>4. The user clicks save</li> <li>5. The page of the students list loads</li> </ol>
<b>Alternative flow 1</b>	The user has the option to return to the list without making any changes by clicking the “return” button
<b>Alternative flow 2</b>	
<b>Post conditions</b>	The student grades get updated

## 4 Design

---

### 4.1 Architecture

---



The config package (Security class) is being called from the system (springboot), thus having no relation with the other packages.

## 4.2 Design

---

Class Name: CourseApplicationSecurityConfig	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ The security of the application</li> <li>▪ Redirecting the user to the log in page</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ None (we think)</li> </ul>

Class Name: CoursesMgtAppController	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Creating the link between the front end (html) and the services</li> <li>▪ Connecting the functions of the app to the UI</li> <li>▪ Connecting functions such as delete from the front end directly to the data base (through the service,dao)</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Course class (model pckg)</li> <li>▪ Student class (model pckg)</li> <li>▪ CourseService (service pckg)</li> <li>▪ StudentService (service pckg)</li> </ul>

Class Name: CourseDAO	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Interface. Declares methods that will be used with the JPA to connect the service to the table courses in the database</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Course (model pckg)</li> </ul>

Class Name: CourseDAOImp	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Do we need it?</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ None (?)</li> </ul>

Class Name: StudentDAO
------------------------

<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Interface. Declares methods that will be used with the JPA to connect the service to the table students in the database</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Student (model pkg)</li> </ul>
--	--

<b>Class Name: StudentDAOImp</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Do we need it?</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ None (?)</li> </ul>

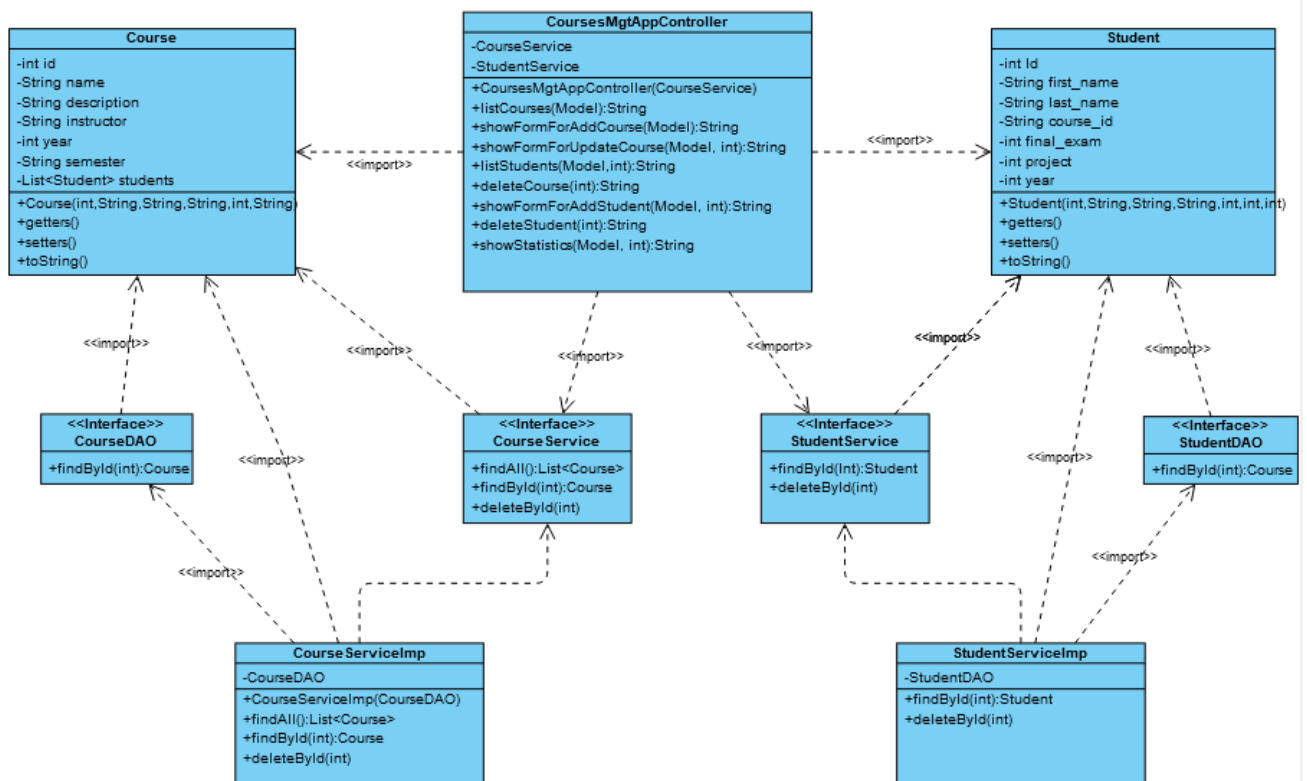
<b>Class Name: Student</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Match the table of the database to fields that can be used later</li> <li>▪ Create get methods that will be used later</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ None</li> </ul>
<b>Class Name: Course</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Match the table of the database to fields that can be used later</li> <li>▪ Create get methods that will be used later</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ None</li> </ul>

<b>Class Name: CourseService</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Interface. Declares the methods that will be implemented</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Course (model pkg)</li> </ul>

<b>Class Name: CourseServiceImp</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Implements the methods that will be used in the controller to access the data from the database</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ CourseDAO (dao pkg)</li> <li>▪ Course (model pkg)</li> <li>▪ StatisticStrategy (statistics pck)</li> </ul>

<b>Class Name: StudentService</b>	
<b>Responsibilities:</b>	<b>Collaborations:</b>
<ul style="list-style-type: none"> <li>Interface. Declares the methods that will be implemented</li> </ul>	<ul style="list-style-type: none"> <li>Student (model pkg)</li> </ul>

<b>Class Name: StudentServiceImpl</b>	
<b>Responsibilities:</b>	<b>Collaborations:</b>
<ul style="list-style-type: none"> <li>Implements the methods that will be used in the controller to access the data from the database</li> </ul>	<ul style="list-style-type: none"> <li>StudentDAO (dao pkg)</li> <li>Student (model pkg)</li> </ul>



Σημείωση: Έγιναν μερικές αλλαγές της τελευταίας στιγμής που δεν προλάβουμε να καταγράψουμε στο report.