
ProfessorWebApp

Sprint Report

DEVELOPERS TEAM

KONSTANTINOS MOUROUSIDIS 4114

ALEXANDROS NIKOLAOU 4126

MICHAIL TSAPANIDIS 4186

VERSIONS HISTORY

Date	Version	Description	Author
25 FEB - 15 APRIL	1.1	User stories first implementation to be functional and the first approach to user interface	Developers Team
15 APRIL - 5 MAY	1.2	User interface and application functionality improvements	Developers Team
5 MAY – 18 MAY	2.0	Final version of the application	Developers Team

1 Introduction

Our target for this application is the management of courses, students and their respective grades from a professor through a web application.

1.1 Document Structure

The rest of this document is structured as follows. Section 2 describes out Scrum team and specifies the 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

Product Owner	UNIVERSITY OF IOANNINA
Scrum Master	APOSTOLOS ZARRAS
Development Team	KONSTANTINOS MOUROUSIDIS ALEXANDROS NIKOLAOU MICHAIL TSAPANIDIS

2.2 Sprints

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

Sprint No	Begin Date	End Date	Number weeks	of	User stories
1	25/2/22	4/3/22	1		US1
2	4/3/22	11/3/22	1		US2,US3
3	11/3/22	18/3/22	1		US4,US5

4	18/3/22	25/3/22	1	US6,US7
5	25/3/22	1/4/22	1	US8,US9
6	1/4/22	8/4/22	1	US10,US11
7	8/4/22	15/4/22	1	US12

3 Use Cases

3.1 <Register>

Use case ID	1
Actors	User
Main flow of events	<ol style="list-style-type: none"> 1. Use case starts when the user presses the register button. 2. User needs to fill his/her credentials in the form. <ol style="list-style-type: none"> 2.1. E-mail 2.2. Password 2.3. First name 2.4. Last name 3. User presses the register button and his/her registration is complete.
Alternative flow 1	If the user does not fill all the required fields in the form, a relative message is displayed.
Alternative flow 2	If the user enters an e-mail or full name (first and last name) address that is already been registered, a respective message is displayed, and the registration will not be completed
Post conditions	User's registration is completed, and he is being redirected to the login page.

3.2 <Log in>

Use case ID	2
Actors	User
Pre conditions	User must be registered in the application.
Main flow of events	<ol style="list-style-type: none">1. Use case starts when user enters the login2. User must enter his login credentials (e-mail and password).3. User presses the login button.
Alternative flow 1	If user enters credentials that are not matching with a registered user, an error message is being displayed.
Post conditions	User logs in the application and the list of his/her courses shows up.

3.3 <Browse courses>

Use case ID	3
Actors	User
Pre conditions	A user must have been logged in
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user has logged in the application.2. The user can see his courses and perform the following actions.<ol style="list-style-type: none">2.1. Edit.2.2. Delete.2.3. Browse Students.2.4. Grades.2.5. Statistics.2.6. Add Course.
Post conditions	Depending of the action that the user chooses, he is redirected to the corresponding page

3.4 <Edit Course>

Use case ID	4
Actors	User
Pre conditions	User has selected the "Edit" action from the "Browse Courses" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user enters the "Edit Course" page2. The user can edit the following fields of a course<ol style="list-style-type: none">2.1. Project percentage2.2. Description of course3. User saves his modifications with the "Save" button or discards them with the "Cancel"
Alternative flow 1	If the "Project Percentage" field value is greater than one (1), user is not being able to proceed to save the modifications
Post conditions	<ol style="list-style-type: none">1. The modifications are saved and user is being redirected to "Browse Courses" page2. If the user presses the "Cancel" button, modifications are discarded and he is being redirected to "Browse Courses" page.

3.5 <Delete Course>

Use case ID	5
Actors	User
Pre conditions	User has selected the "Delete" action from "Browse Courses" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when user presses the "Delete" button2. The course is being deleted from the database
Post conditions	Course is deleted from the database and the course list is refreshed

3.6 <Add course>

Use case ID	6
Actors	User
Pre conditions	User has selected "Add course" from "Browse Courses" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the "Add course" button2. User is being redirected to the respective page for adding a course to the database3. User fills out:<ol style="list-style-type: none">3.1. Course name-title3.2. Course description4. User selects:<ol style="list-style-type: none">4.1. Save if he wants to save the course in the4.2. Cancel if he wants to discard the course insertion in the database5. User saves the new course or returns to the "Browse Courses" page
Alternative flow 1	If the user fills a course name that already exists, the new course can not be saved and a relevant message is
Post conditions	Database gets updated with the new course and user returns to "Browse Courses" page and views the updated course

3.7 <Show Statistics>

Use case ID	7
Actors	User
Pre conditions	User has selected the "Statistics" action from "Browse Courses" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the "Statistics" button from "Browse Course" page2. A table with the course's statistics is displayed

3.8 <Grades>

Use case ID	8
Actors	User
Pre conditions	User has selected the “Grades” action from “Browse Courses” page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the “Grades” button from “Browse Courses” page2. A table containing the students and their grades for the course is displayed3. User has the choice to edit the grades of each student by pressing the “Edit Grades” button
Post conditions	User is redirected to the “Edit Grades” page if he chooses so, or he is staying in “Grades” page

3.9 <Browse Students>

Use case ID	9
Actors	User
Pre conditions	User has selected the “Browse Students” action from “Browse Courses” page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the “Browse Students” button from “Browse Courses” page.2. A table with the students who participate in the course is displayed3. User can view the participating students and has the following choices:<ol style="list-style-type: none">3.1. Edit3.2. Delete3.3. Add a student
Post conditions	User is being redirected to the respective page depending on his choice of action

3.10 <Edit Student>

Use case ID	10
Actors	User
Pre conditions	User has selected the "Edit" action from "Browse Students" page.
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the "Edit" button from "Browse Students" page.2. User is able to edit the following fields:<ol style="list-style-type: none">2.1. First name2.2. Last name2.3. email
Post conditions	<ol style="list-style-type: none">1. If the user chooses "Save", the database is updated with the new student credentials and he is redirected to "Browse Students" page where he can see the updated student list.2. If the user chooses "Cancel", the database is not updated and he is redirected to "Browse Students" page.

3.11 <Delete Student>

Use case ID	11
Actors	User
Pre conditions	User has selected the "Delete" action from "Browse Courses" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the "Delete" button from "Browse Students" page.2. Student gets deleted from the database
Post conditions	Student is deleted from the database and the user is redirected to "Browse Students" page.

3.12 <Add Student>

Use case ID	12
Actors	User
Pre conditions	User has selected the "Add a student" action from "Browse Students" page
Main flow of events	<ol style="list-style-type: none">1. Use case starts when the user presses the "Add a student" button.2. User is redirected to the student registration page.3. User fills out:<ol style="list-style-type: none">3.1. Student's first name3.2. Student's last name3.3. Student's e-mail4. User selects:<ol style="list-style-type: none">4.1. Add if he wants to save the new student in the database4.2. Cancel if he wants to cancel the student registration5. User saves the new student or returns to "Browse Students" page.
Alternative flow 1	If the user fills a full name (first and last name) or e-mail that has already been registered, no registration is being made and a relative message is displayed.
Post conditions	<ol style="list-style-type: none">1. If the user presses the "Add" button, database is updated with the new student, he returns to "Browse Students" page and sees the update student list2. If the user presses the "Cancel" button, database is not updated and he is redirected to "Browse Students" page.

3.13 <Home Button >

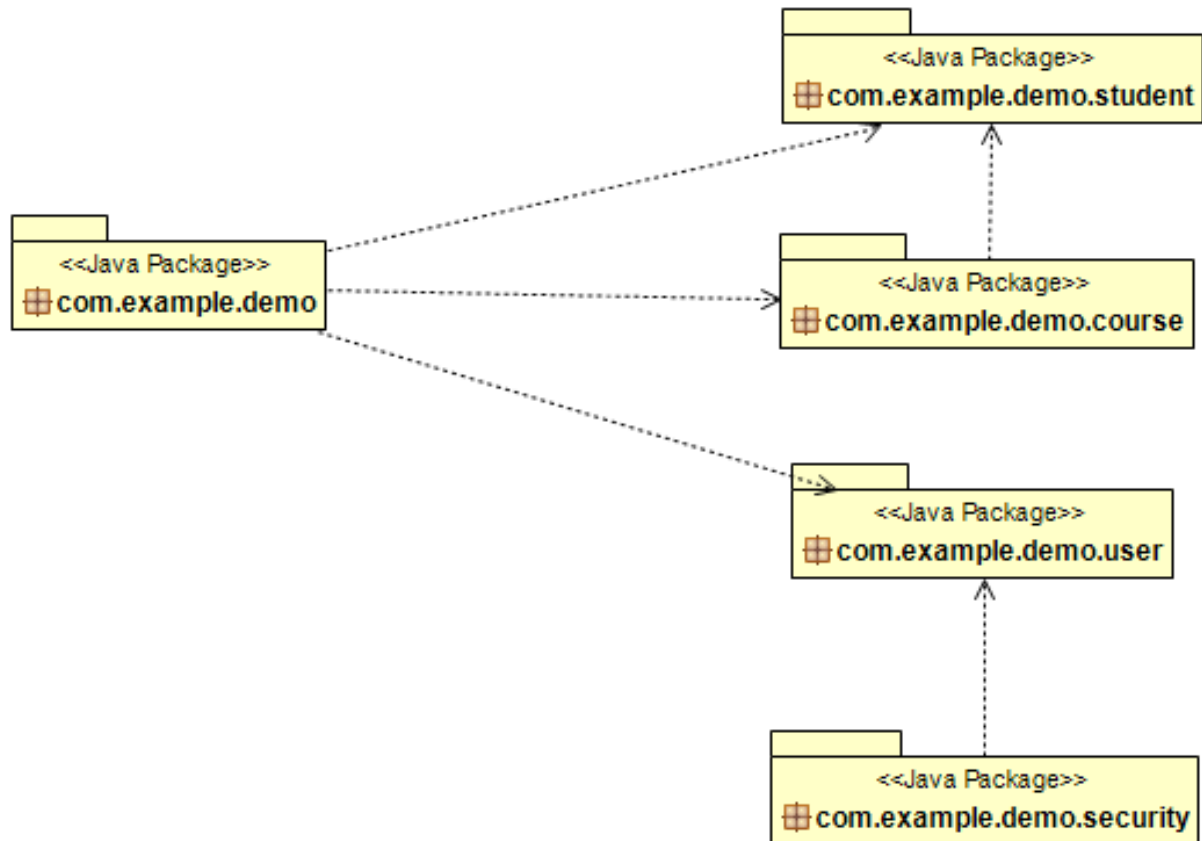
Use case ID	13
Actors	User
Pre conditions	User has selected the “Professor Class Web” action (top left corner) from any page
Main flow of events	Use case starts when the user presses the “Professor Class Web” from any page
Post conditions	User is redirected to “Browse Courses” page

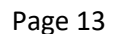
3.14 <Log Out Button >

Use case ID	14
Actors	User
Pre conditions	User has selected the “Log Out” action (top right corner) from any page.
Main flow of events	Use case starts when the user presses the “Log Out” button from any page
Post conditions	User is logged out of the application

4 Design

4.1 Architecture





Class Name: AppController	
Responsibilities: <ul style="list-style-type: none"> ▪ Communicating between front-end and back-end software ▪ Handling and passing arguments between to and from HTML pages ▪ Handling the mapping of HTML pages 	Collaborations: <ul style="list-style-type: none"> ▪ All the repository Classes ▪ All the service Classes ▪ Entities

Class Name: CodeJavaAppApplication	
Responsibilities: <ul style="list-style-type: none"> ▪ Main Class of the application ▪ Starts the application ▪ Boots Spring Framework 	Collaborations: <ul style="list-style-type: none"> ▪ Spring Framework

Class Name: User	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates a table with user entities ▪ Handles the information about user entities 	Collaborations:

Class Name: UserRepository	
Responsibilities: <ul style="list-style-type: none"> ▪ Extends JPA repository ▪ Handles queries for necessary info from the database 	Collaborations: <ul style="list-style-type: none"> ▪ User

Class Name: CustomUserDetails	
Responsibilities: <ul style="list-style-type: none"> ▪ Authorizes the user 	Collaborations: <ul style="list-style-type: none"> ▪ User

Class Name: CustomUserDetailsService	
Responsibilities: <ul style="list-style-type: none"> ▪ Communication with the user repository ▪ Methods to access information we need from user repository 	Collaborations: <ul style="list-style-type: none"> ▪ User ▪ CustomUserDetails ▪ UserRepository

Class Name: Students	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates a table with student entities ▪ Handles the information about student entities 	Collaborations:

Class Name: StudentsRepository	
Responsibilities: <ul style="list-style-type: none"> ▪ Extends JPA repository ▪ Handles queries for necessary info from the database 	Collaborations: <ul style="list-style-type: none"> ▪ Students

Class Name: StudentService	
Responsibilities: <ul style="list-style-type: none"> ▪ Communication with the students repository ▪ Methods to access information we need from students repository 	Collaborations: <ul style="list-style-type: none"> ▪ Students ▪ StudentsRepository

Class Name: Course	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates a table with course entities ▪ Handles the information about course entities 	Collaborations:

Class Name: CoursesRepository	
Responsibilities: <ul style="list-style-type: none"> ▪ Extends JPA repository ▪ Handles queries for necessary info from the database 	Collaborations: <ul style="list-style-type: none"> ▪ Course

Class Name: CourseService	
Responsibilities: <ul style="list-style-type: none"> ▪ Communication with the students repository ▪ Methods to access information we need from students repository 	Collaborations: <ul style="list-style-type: none"> ▪ Course ▪ CoursesRepository

Class Name: WebSecurityConfig	
Responsibilities: <ul style="list-style-type: none"> ▪ Handles the security of the application ▪ Authenticates the user 	Collaborations: <ul style="list-style-type: none"> ▪ CustomUserDetailsService

Class Name: PasswordEncoder	
Responsibilities: <ul style="list-style-type: none"> ▪ Encodes password 	Collaborations:

Thank you.

Developers Team

Ioannina, 17/05/2022