
CoursesManagementApp

Sprint Report

The 10x'ers

Koureas Athanasios, 4392

Athanasios Papapostolou, 4147

Konstantinos Georgiou, 4333

VERSIONS HISTORY

Date	Version	Description	Author
26/3/2022	v0.1	Tooling and basic requirements	Koureas Athanasios
2/4/2022	v1.0	-----	Koureas Athanasios

1 Introduction

This document provides information concerning the **1st** sprint of the project.

2 Scrum team and Sprint Backlog

2.1 Scrum team

Product Owner	Koureas Athanasios
Scrum Master	Konstantinos Georgiou
Development Team	Konstantinos Georgiou, Papapostolou Athanasios, Koureas Athanasios

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	19/3/2022	26/3/2022	1	NF-1
2	26/3/2022	2/4/2022	1	US-2,US-3,US-4
3	2/4/2022	9/4/2022	1	US-5,US-6,US-7,US-8
4	9/4/2022	16/4/2022	1	
5	16/4/2022	23/4/2022	1	

2.3 Sprint Backlog

Requirement Id	Which Sprint
NF1	Sprint No1
US-2,US-3,US-4	Sprint No2
US-5,US-6,US-7,US-8	Sprint No3

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>

Use case ID	Register Grades
Data	Grades id Project grade Exam grade Semester
Actors	Professor
Pre Conditions	1. Course object should exist 2. Student list should have at least one item
Main flow of events	1. Professor issues the "Add Grades" command 2. Professor fills in the data textfields accordingly 3. Professor presses the "OK" button 4. System refreshes the Student item view
Fields not filled	1. System notifies professor that a text field is not filled with data 2. System does not proceed and prompts professor to fill in the remaining fields
Post conditions	1. Grades of student should get added for the current semester

3.2 <Use Case 2>

Use case ID	Professor Register
Data	Username Password
Actors	Professor
Pre Conditions	<>
Main flow of events	<ol style="list-style-type: none">1. Professor fills in the username and password textfields accordingly2. Professor presses the "Register" button3. System saves registration data in DBMS4. System execute the "Login" use case with same data
Field not filled	<ol style="list-style-type: none">1. System notifies professor that a text field is not filled with data2. System does not proceed and prompts professor to fill in the remaining fields
Invalid Data Student	<ol style="list-style-type: none">1. System finds an error in the provided data2. System does not proceed and prompts professor to fill in the remaining fields
User Exists	<ol style="list-style-type: none">1. System notifies professor that the registration he is trying to create already exists2. System executes "Login" use case with same data
Post conditions	<ol style="list-style-type: none">1. List of professor is updated with the new item

3.3 <Use Case 3>

Use case ID	Professor Login
Data	Username Password
Actors	Professor
Pre Conditions	1. Registration token with according data should exist
Main flow of events	1. Professor fills in the username and password textfields accordingly 2. Professor presses the “Login” button 3. System views the main page
Fields not filled	1. System notifies professor that a text field is not filled with data 2. System does not proceed and prompts professor to fill in the remaining fields
Post conditions	<>

4 Design

4.1 Architecture

<Specify the overall architecture for this release in terms of a **UML package diagram**.>

4.2 Design

<Specify the detailed design for this release in terms of **UML class diagrams**.>

<Document the classes that are included in this release in terms of CRC cards according to the template that is given below.>

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

▪ ...	▪
-------	--------

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

.....

.....