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**

2.3 Sprint Backlog

Requirement Id	Which Sprint
NF1	**Sprint No1**
US-2,US-3,US-4	**Sprint No2**

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	Browse List
Data	List of courses
Actors	Professor
Main flow of events	1. User presses the 'Browse' button 2. User scrolls through the list of courses
Post conditions	1. User is presented with a list of courses retrieved from a DBMS 2. New courses are presented dynamically in a lazy list fashion

3.2 <Use Case 2>

Use case ID	Add Course
Data	List of courses Course id Course name Syllabus Year Semester
Actors	Professor
Main flow of events	1. User presses the 'Add Course' button on top of the list view 2. User inputs textfields of new course data 3. User confirms data and presses the 'Add' button 4. List of courses is reloaded
Post conditions	1. User is presented with a new course view with empty textfields with placeholders 2. Textfields save data 3. Data is saved according to a DBMS Schema 4. List of courses previews the updated list with the newly added object

3.3 <Use Case 3>

Use case ID	US4 Remove Course
Data	List of courses Course id
Actors	Professor
Main flow of events	1. User presses the 'Remove Course' button next to a course item 2. User presses the 'OK' button 3. List of courses is reloaded (reactively)
Post conditions	1. List item is sliced over and reveals the 'OK' button 2. Course id is sent to the DBMS to delete 3. List of courses previews the updated list without the deleted object

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: ▪ ... ▪ .... ▪ ...	**Collaborations:** ▪ ... ▪ .... ▪ ....

Class Name:	
Responsibilities: ▪ ... ▪ .... ▪ ...	**Collaborations:** ▪ ... ▪ .... ▪ ....

.....

.....