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**	

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	Browse Student List
Data	<>
Actors	Professor
Pre Conditions	1. Should be viewing a particular course item
Main flow of	1. Professor issues “Browse Students” command

events	2. System selects students list from the DBMS 3. User is presented with list-view of all students enrolled in the course
No Students Error	1. System delivers error message “Oops, no students enrolled yet to user”
Post conditions	<>

3.2 <Use Case 2>

Use case ID	Add Student Details
Data	Student id Student name Year of registration Semester
Actors	Professor
Pre Conditions	<>
Main flow of events	1. Professor issues “Add details” command 2. Professor fills in the data textfields accordingly 3. Professor presses the “OK” button 4. System repeats the “Browse Student List” use case
Field 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
Exiting Student	1. System notifies the professor that the student attempting to add exists in the list 2. System repeats “Add Student Details”
Post conditions	1. List of students is updated with the new item

3.3 <Use Case 3>

Use case ID	Remove Student
Data	Student item
Actors	Professor
Pre Conditions	1. Should be viewing a particular course item 2. Should exist at least one student item
Main flow of events	1. Professor issues “Delete Student” command 2.System issues prompts professor with an “Are you Sure” alert 3. Professor confirms the deletion of an item 4. List of students removes the selected item
Post conditions	1. List of students is updated without the recently removed student item

3.4 <Use Case 4>

Use case ID	Update Student Details
Data	Student id Student name Year of Registration Semester
Actors	Professor
Pre Conditions	Student item we want to update should exist
Main flow of events	1. Professor issues “Update Student” command 2.System prompts professor with form 3. Professor fills in the data textfields accordingly 4. Professor presses the “OK” button 5. System repeats the “Browse Student List” use case
Post conditions	1. List of students is updated with the new student data

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:** ▪ ... ▪ .... ▪ ....

.....

.....