Diploma Projects Management App

Sprint Report

Vasilios Valsamidis , 4321

Dimitrios Giannitsakis, 4338

Andreas Birmpilis, 4436

Athanasios Kalyviotis, 4607

VERSIONS HISTORY

Date	Version	Description	Author
26/04/2023	v.1.0	Inserted basic team info	A. Kalyviotis
03/05/2023	v.1.1	Inserted use cases	D. Giannitsakis
20/05/2023	v.1.2	Inserted CRC cards	A. Kalyviotis

1 Introduction

This document provides information concerning the last sprint of the project.

1.1 Purpose

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

2.1 Scrum team

Product Owner	Thanasis Kalyviotis
Scrum Master	Vasilis Valsamidis
Development Team	Andreas Birmpilis, Dimitrios Giannitsakis

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	**07/4/2023**	**14/4/2023**	**1**	**Basic User Sign-in and Registration**
2	**28/4/2023**	**5/5/2023**	**1**	**Student and Professor Profile Information Implementation**
3	**5/5/2023**	**12/5/2023**	**1**	**Diploma thesis subjects list/detail views and CRUD implementation. Apply to thesis subjects functionality for Students.**
4	**12/5/2023**	**17/5/2023**	**1**	**Assignment and evaluation of thesis subjects**

3 Use Cases

3.1 <Use Case 1>

Use case Name	CreateAccount
Actors	User
Pre conditions	None
Main flow of events	1. The use case starts when the user clicks on the Register button 2. The system shows a user registration form 3. The user enters their username, password, email address 4. The user clicks on the “Choose a Role” and selects a role 4.1. If they choose Student, they set their personal profile information and the system registers them as a Student 4.1.1. The system displays to the student to complete his years of studies, course remaining and average grade 4.2. If they choose Professor, they sets their personal profile information and the system registers them as a Professor 4.2.1 The system displays to the professor to complete his speciality 5. The user clicks “Sign up” button
Alternative flow 1	1. If the user uses username and email which already exist, the system will not allow the user to register 2. If the student uses decimal numbers for the year of studies and courses remaining, the system will display a message that the values are not valid
Alternative flow 2	None
Post conditions	The user has created their account and has chosen their role (Student or Professor). The form data has been stored in the database.

3.2 <Use Case 2>

Use case Name	LoginToTheApplication
Actors	User
Pre conditions	The user must have created his account
Main flow of events	1 . The use case starts when the user clicks on the Login button 2 . The user enters his username and password 3 . The user clicks Sign in
Alternative flow 1	If the user types a username which does not exist, the system will appear "Username or password is invalid"
Alternative flow 2	If the user types a password which does not exist, the system will appear "Username or password is invalid"
Post conditions	The user is connected to the application

3.3 <Use Case 3>

Use case Name	AccessThesisList
Actors	Student
Pre conditions	The student must be connected to the application
Main flow of events	1. The use case starts when the student hits the "View Diploma Thesis Subjects" Button 2.The system shows a list with all the diploma thesis subject
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The student has access to a list of available diploma thesis subject

3.4 <Use Case 4>

Use case Name	ViewDescriptionOfDiplomaThesisSubject
Actors	Student, Professor
Pre conditions	The student or professor must have selected a list of available diploma thesis subject
Main flow of events	1. The use case starts when the student clicks on one diploma thesis subject 2 . The system redirects to a new page with details about the selected diploma thesis subject appears (name's professor, objectives, number of applications)
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The student can see a more detailed description of a diploma thesis subject

3.5 <Use Case 5>

Use case Name	ApplyDiplomaThesis
Actors	Student
Pre conditions	The student must have access to a list of available thesis subjects
Main flow of events	1.The use case starts when the student clicks “Apply” Button 2. The system shows a confirmation message 3. The system increases the number of applications
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The student can apply for a diploma thesis subject

3.6 <Use Case 6>

Use case Name	CancelApplication
Actors	Student
Pre conditions	The student must press ‘Apply’ button for a diploma thesis subject
Main flow of events	1 . The use starts when the user clicks “Cancel Application” button 2 . The system will decrease the number of applications
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The student has cancelled the application for a thesis subject

3.7 <Use Case 7>

Use case Name	AddNewDiploma
Actors	Professor
Pre conditions	The user must be connected as a professor and must have hit the “View Diploma thesis subject” Button
Main flow of events	1. The use case starts when the professor hits button ‘Add’ Thesis” from the list view interface 2. The system shows a form where the professor can write the title of the new thesis subject and the objective 3. The user clicks on the ’Save’ button 4. The system redirects the professor back to the list of available thesis subjects with the new diploma thesis subject

Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has added the new subject

3.8 <Use Case 8>

Use case Name	DeleteDiplomaThesis
Actors	Professor
Pre conditions	The user must be connected as a professor and have at least one thesis subject created
Main flow of events	1. The use case starts when the professor hits the 'Delete' button next to each thesis subject 2. The system shows a confirmation message 3. If the professor clicks "Cancel" 3.1 The thesis subject will not be deleted 4.If the professor clicks "Yes,I am sure" 4.1 The thesis subject will be deleted 5 . The system redirects the professor back to the list of available thesis subjects
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has deleted a diploma thesis subject from his list

3.9 <Use Case 9>

Use case Name	UpdateDiplomaThesis
Actors	Professor
Pre conditions	The user must be connected as a professor and have at least one thesis subject created
Main flow of events	1. The use case starts when the professor hits the 'Update' button next to each thesis subject 2. The professor can change tittle and objectives 3 .The professor clicks on the 'Save' button 4. The system redirects the professor back to the list of available thesis subjects
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has updated a diploma thesis subject from his list

3.10 <Use Case 10>

Use case Name	ViewTheListOfApplications
Actors	Professor
Pre conditions	The user must be connected as a professor and have at least one thesis subject created
Main flow of events	1. The use case starts when the professor clicks on one diploma thesis subject 2. The system shows a list of applicants and all their details
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has access to a list of applications from the students who want to take over a diploma thesis subject

3.11 <Use Case 11>

Use case Name	AssignDiplomaThesisSubject
Actors	Professor
Pre conditions	The user must be connected as a professor and have at least one thesis subject created
Main flow of events	1 . The use case starts when the professor hits the “To Assignment page” button in the detail view of the thesis 2 . The professor selects a strategy from the dropdown of the new page that loads. 3 . For each interested student, if the professor clicks “Random choice” 3.1 The system will assign a diploma thesis subject in a random way 4 . If the professor clicks “Best average courses grade” 4.1 The system will assign a diploma thesis subject to a student with the best average courses grade 5 . If the professor clicks “ The fewest remaining courses”

	5.1 The system will assign a diploma thesis subject to a student with the fewest remaining courses for graduation 6. If the professor clicks “Average courses grade greater than a threshold and number of remaining courses less than a threshold “ 6.1. The system will assign a diploma thesis subject to a student with average courses grade greater than a given threshold Th1 and number of remaining courses for graduation less than a given threshold Th2 7 . The professor hits the “Assign” button, the system deletes all the applicants who have hits “Apply” for the specific subject 8 . The system shows all the details for the student who is selected
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has assigned a diploma thesis subject to one of the students and the student can start working on the project

3.12 <Use Case 12>

Use case Name	CancelDiplomaThesisSubject
Actors	Professor
Pre conditions	The professor must have assigned a diploma thesis subject to one of the students
Main flow of events	1 . The use case starts when the professor hits the “Cancel” Button 2 . The system deletes all the students who hit “Apply” for the specific subject
Alternative flow 1	None
Alternative flow 2	None
Post conditions	There are no applicants to this thesis subject

3.13 <Use Case 13>

Use case Name	AccessToTheAssigneeList
Actors	Professor
Pre conditions	The professor must have assigned the diploma thesis subjects to a student
Main flow of events	1 . The use case starts when the professor clicks for each diploma thesis subject 2 . The system redirects to a detail-view page with details about the thesis and displays the message “This thesis has been assigned to a student” 3 . If the professor hits “To Assignmet Page “, the system redirects to a new page with all the details of the assigned student
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has access to the list of assigned diploma thesis project that he supervises

3.14 <Use Case 14>

Use case Name	EvaluateTheWork
Actors	Professor
Pre conditions	The professor must have assigned the diploma thesis subjects to a student
Main flow of events	1 . The use case starts when the professor clicks “To Evaluation Page” in the assignment view 2. The system redirects the professor to a new page with all details for the assigned student and one form for the implementation grade, report grade and presentation grade which the professor must pr 3. The professor can select the evaluation formula in order to calculate the total grade 4. The professor clicks on the “Evaluate” Button 5. The system displays a confirmation message 6. If the professor clicks “Yes” 6.1. The system refreshes the page and displays the grades of the student’s implementation of the diploma thesis 7 . If the user clicks “Cancel” 7.1 The system aborts the grade saving procedure
Alternative flow 1	None
Alternative flow 2	None
Post conditions	The professor has evaluated the student the work’s student

4 Design

4.1 Architecture

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

4.2 Design

Class Name: AuthController	
Responsibilities:	**Collaborations:**
▪ Handle the HTTP requests for the user authentication (registration and sign in)	▪ UserService ▪ StudentService ▪ ProfessorService ▪ User ▪ Student ▪ Professor

Class Name: HomeController	
Responsibilities:	**Collaborations:**
▪ Handle the HTTP requests for the rendering of the about us page	

Class Name: ProfessorController	
Responsibilities:	**Collaborations:**
▪ Handle the HTTP requests for the professor profile view rendering	▪ Professor ▪ User ▪ ProfessorService ▪ UserService

Class Name: StudentController	
Responsibilities:	**Collaborations:**
▪ Handle the HTTP requests for the student profile view rendering	▪ Student ▪ User ▪ UserService ▪ StudentService

Class Name: ThesisController	
Responsibilities:	**Collaborations:**
▪ Handle the HTTP requests for the thesis creation, update and deletion ▪ Handle the HTTP requests for the students’ assignment to a thesis and the application undo ▪ Handles the HTTP requests for the list view of the thesis subjects and the detail view ▪ Handles the HTTP requests for the assignment of the thesis subject to a student and the evaluation of assignments by the professors for the implementation	▪ Application ▪ Assignment ▪ BestAverageSelectionStrategy ▪ Evaluation ▪ EvaluationFormula ▪ FewestCoursesSelectionStrategy ▪ Professor ▪ RandomSelectionStrategy ▪ SelectionStrategy ▪ StandardEvaluationFormula ▪ Student ▪ Thesis ▪ ThresholdSelectionStrategy ▪ User ▪ ApplicationService ▪ AssignmentService ▪ EvaluationnService ▪ ProfessorService ▪ StudentService ▪ ThesisService ▪ UserService

Class Name: UserForm	
Responsibilities: ▪ Represents the User creation form	**Collaborations:** ▪ Role

Class Name: ApplicationDAO	
Responsibilities: ▪ Acts as an intermediate between the Application service and the database – takes care of the querying to the database and the handling of the student’s applications	**Collaborations:** ▪ Application ▪ Professor ▪ Student ▪ User

Class Name: AssignmentDAO	
Responsibilities: ▪ Acts as an intermediate between the Assignment service and the database – takes care of the querying to the database and the handling of the thesis subjects’ assignments	**Collaborations:** ▪ Assignment ▪ Student ▪ Thesis

Class Name: EvaluationDAO	
Responsibilities: ▪ Acts as an intermediate between the Evaluation service and the database – takes care of the querying to the database and the handling of the students’ implementation evaluations	**Collaborations:** ▪ Evaluation

Class Name: ProfessorDAO	
Responsibilities: ▪ Acts as an intermediate between the Professor service and the database – takes care of the querying to the database and the handling of the professor profile information	**Collaborations:** ▪ Professor ▪ User

Class Name: StudentDAO	
Responsibilities: ▪ Acts as an intermediate between the Student service and the database – takes care of the querying to the database and the handling of the student profile information	**Collaborations:** ▪ Student ▪ User

Class Name: ThesisDAO	
Responsibilities:	**Collaborations:**
▪ Acts as an intermediate between the Thesis service and the database – takes care of the querying to the database and the handling of the thesis information and details	▪ Thesis

Class Name: UserDAO	
Responsibilities:	**Collaborations:**
▪ Acts as an intermediate between the User service and the database – takes care of the querying to the database and the handling of the users as an authorization element of the app	▪ User

Class Name: Application	
Responsibilities:	**Collaborations:**
▪ Represents the student application	▪ Student ▪ Thesis

Class Name: Assignment	
Responsibilities: ▪ Represents the assignment of a thesis subject to a student	**Collaborations:** ▪ Student ▪ Thesis

Class Name: Evaluation	
Responsibilities: ▪ Represents the evaluation of a student’s implementation of a thesis subject from the professor	**Collaborations:** ▪ Assignment

Class Name: FewestCoursesSelectionStrategy	
Responsibilities: ▪ Implements the selection strategy interface	**Collaborations:** ▪ Student

Class Name: Professor	
Responsibilities: ▪ Represents the professor information	**Collaborations:** ▪ User

Class Name: RandomSelectionStrategy	
Responsibilities: ▪ Implements the selection strategy interface	**Collaborations:** ▪ Student ▪ SelectionStrategy

Class Name: SelectionStrategy	
Responsibilities: ▪ Defines an interface for the strategy of selecting a candidate student for being assigned to a diploma thesis subject	**Collaborations:** ▪ Student

Class Name: Student	
Responsibilities: ▪ Represents the student information	**Collaborations:** ▪ User

Class Name: Thesis	
Responsibilities: ▪ Represents the thesis information	**Collaborations:** ▪ Professor

Class Name: ThresholdSelectionStrategy	
Responsibilities: ▪ Implements the selection strategy interface	**Collaborations:** ▪ Student

Class Name: ApplicationService	
Responsibilities:	**Collaborations:**
▪ Provides	▪ Application ▪ Student

Class Name: ApplicationServiceImpl	
Responsibilities:	**Collaborations:**
▪ Implements the Application Service	▪ ApplicationService ▪ ApplicationDAO ▪ Application ▪ Student

Class Name: AssignmentService	
Responsibilities:	**Collaborations:**
▪ Combines different DAO methods to provide the ThesisController with the requested data and delegates calls to commit assignments to the database	▪ Assignment ▪ Student

Class Name: AssignmentServiceImpl	
Responsibilities: ▪ Implements the Assignment Service	**Collaborations:** ▪ AssignmentDAO ▪ Assignment ▪ Student ▪ AssignmentService

Class Name: EvaluationService	
Responsibilities: ▪ Combines different DAO methods to provide the ThesisController with the requested data and delegates calls to commit evaluations to the database	**Collaborations:** ▪ Evaluation

Class Name: EvaluationServiceImpl	
Responsibilities: ▪ Implements the Evaluation Service	**Collaborations:** ▪ EvaluationDAO ▪ Assignment ▪ Evaluation ▪ EvaluationService

Class Name: ProfessorService	
Responsibilities: ▪ Combines different DAO methods to provide the ThesisController and the ProfessorController with the requested data and delegates calls to commit professor profile information to the database	**Collaborations:** ▪ Professor ▪ User ▪

Class Name: ProfessorServiceImpl	
Responsibilities: ▪ Implements the Professor Service	**Collaborations:** ▪ ProfessorDAO ▪ Professor ▪ User ▪ ProfessorService

Class Name: StudentService	
Responsibilities: ▪ Combines different DAO methods to provide the ThesisController and the StudentController with the requested data and delegates calls to commit evaluations to the database	**Collaborations:** ▪ Student ▪ User

Class Name: StudentServiceImpl	
Responsibilities:	**Collaborations:**
▪ Implements the Student Service	▪ StudentDAO ▪ User ▪ Student ▪ StudentService

Class Name: ThesisService	
Responsibilities:	**Collaborations:**
▪ Combines different DAO methods to provide the ThesisController with the requested data and delegates calls to commit evaluations to the database	▪ Thesis

Class Name: ThesisServiceImpl	
Responsibilities:	**Collaborations:**
▪ Implements the Thesis Service	▪ ThesisService ▪ ThesisDAO ▪ Thesis

Class Name: UserService	
Responsibilities:	**Collaborations:**
▪ Combines different DAO methods to provide the ThesisController and the AuthController with the requested data and delegates calls to commit evaluations to the database	▪ User

Class Name: UserServiceImpl	
Responsibilities:	**Collaborations:**
▪ Implements the User Service	▪ UserService ▪ UserDetailsService ▪ UserDAO ▪ User