Software System Verification and Validation
2019-2020
Laboratory Assignment 01
www.cs.ubbcluj.ro/~avescan

Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

8th March, 2020

Developed by: Adascalitei Alexandru, Bodea Adonis - 931

Version History

Version	Description of Change	Author	Date
V01	Initial/Modification of document	Adascalitei Alexandru, Bodea Adonis	07.03.2020
V02	Completion of document	Adascalitei Alexandru, Bodea Adonis	08.03.2020

Software System Verification and Validation
2019-2020
Laboratory Assignment 01
www.cs.ubbcluj.ro/~avescan

Contents

Analysis and design Document

1. Functional Requirements

Section/ Requirement ID	Requirement Definition
FR 1.0.	CRUD operations for the Student entity.
FR 2.0.	Create a laboratory theme entity.
FR 3.0.	Update the deadline of existing laboratory theme entity (Available if the current week number is less than or equal to the sum of the week number of the assignment deadline and the number of weeks with which the deadline to be extended).
FR 4.0.	When adding a new laboratory theme, as well as modifying the delivery date of a theme, all students will be notified by email.
FR 4.1.	The application will offer the ability to unsubscribe from these notifications.
FR 5.0.	Create a grade entity for a student at a laboratory theme.
FR 5.1.	A student has exactly one grade for a laboratory theme.
FR 5.2.	Any deduction due to delays in the delivery of a theme is automatically computed.
FR 5.3.	The deduction due to delays is not to be considered if the student presents a medical motivation.
FR 5.4.	The information is saved in a file, with the name of the form “StudentName.txt”, as an entry of the form: "Theme:" ThemeID, “Grade:” Grade of the student for this theme "Delivered in the week:" Number of the week in which the assignment was a delivered "Deadline:" The deadline week of this theme "Feedback:" feedback, suggestions, and explanations in connection with the grade
FR 6.0.	The NameStudent.txt file content will be emailed to the student weekly, with the subject "Feedback laboratory MAP"

Section/ Requirement ID	Requirement Definition
FR 7.0.	Filter entities based on certain criteria.
FR 8.0.	Create reports. Examples: Laboratory grade for each student - the weighted average of grades from the lab topics where the weight share is the number of weeks allocated to the topic The hardest theme – the theme with the smallest average of the grades Students who can enter the exam - average greater than or equal to 4 Students who have delivered all the themes on time

2. Actors

- Teacher

3. Use cases – diagram

3.1. Use case no. 1

Actors: Teacher

Description: Display all students.

Precondition: A valid XML file is given as input source and is loaded into the internal memory, otherwise an error occurs: 'java.io.FileNotFoundException: /Users/alexdarie/Downloads/LabAssiAsseProjectV01/studenti.xml (No such file or directory)'

Postcondition: None.

User action	System response
User opens the application.	
	A list of options is displayed being followed by the application specific prompt.
The indicative value of the option (in this case '11') is typed to the command line and submitted by pressing Enter.	
	A visual feedback is given by presenting the list of existing students. If the input source is empty the prompt will be displayed immediately after the command was submitted.

Exceptions: 'Exception in thread "main" java.lang.NullPointerException' can occur if the xml file is missing mandatory tags. For example the absence of '<Grupa>' from a '<student>' tag can generate this error.

3.2. Use case no. 2

Actors: Teacher

Description: Display all assignments.

Precondition: A valid input source must be specified.

Postcondition: None.

User action	System response
User opens the application.	
	A list of options is displayed being followed by the application specific prompt.

The indicative value of the option (in this case '12') is typed to the command line and submitted by pressing Enter.	
	A visual feedback is given by presenting the list of existing assignments. If the input source is empty the prompt will be displayed immediately after the command was submitted.

Exceptions: 'Exception in thread "main" java.lang.NullPointerException' can occur if the xml file is missing mandatory tags. For example the absence of '<Descriere>' from a '<tema>' tag can generate this error.

3.3. Use case no. 3

Actors: Teacher

Description: Display all grades for a specific student.

Precondition: The student must exist. Also, a valid XML file is given as input source and is loaded into the internal memory, otherwise an error occurs. See use case no. 1 for more details on the error that occurs.

Postcondition: None.

User action	System response
User opens the application.	
	A list of options is displayed being followed by the application specific prompt.
The indicative value of the option (in this case '13') is typed to the command line and submitted by pressing Enter.	
	A visual feedback is given by presenting the list of grades for a specific student. If the student has no grade assigned yet the prompt will be displayed immediately after the command was submitted.

Exceptions: 'Exception in thread "main" java.lang.NullPointerException' can occur if the xml file is missing mandatory tags. For example the absence of '<Feedback>' from a '<nota>' tag can generate this error.

4. Analysis

4.1. Entities

Grade, Student, Assignment

4.2. Relations between entities

An assignment has a grade. A Student can have many assignments and an assignment can be assigned to more than one student.

4.3. Attributes

Grade: id, grade, deadline, feedback

Student: id, name, group

Assignment: id, description, deadline, startline

4.4. System behavior

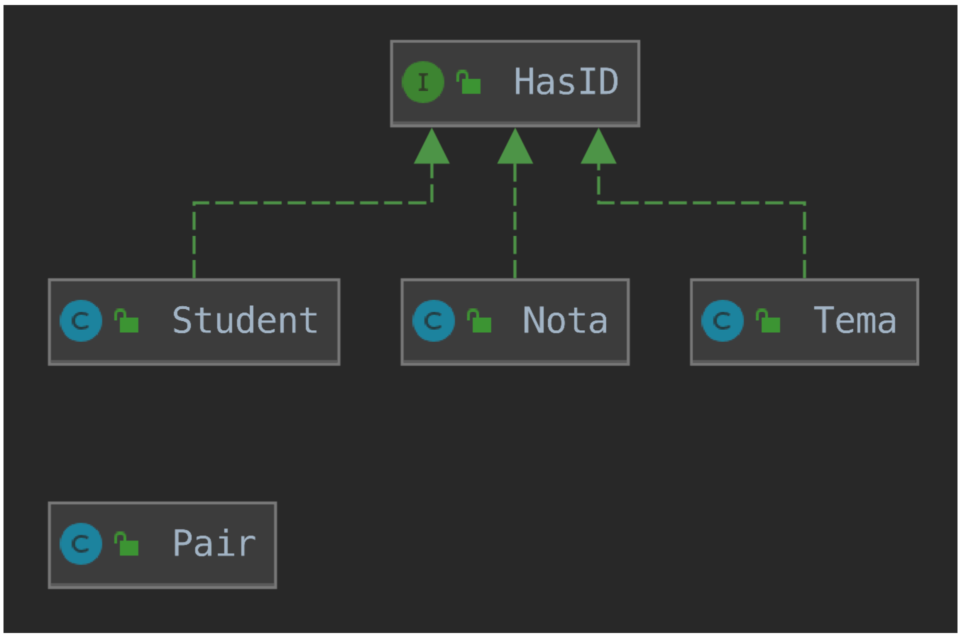
All use cases described above are not intended to be integrated in a larger application. In fact, integrating this application in any other existing systems is considered a challenge. More on this can be found in the DesignPhaseDefectsChecklist and ProgramCodingPhaseDefectsChecklist document. The application will be used as a standalone software product.

4.5. System events

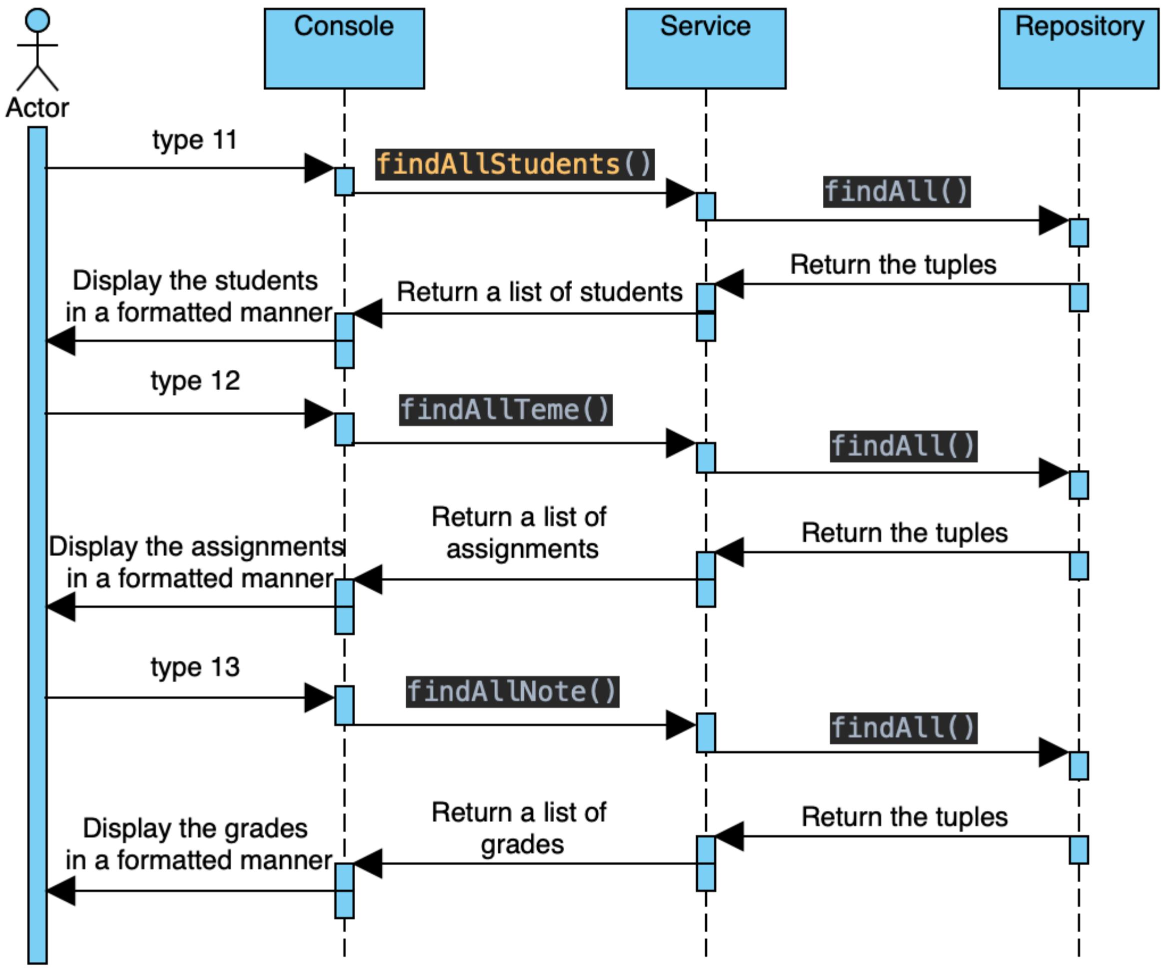
All the events or activities that take place in the given application are either triggered by command line input sequences or as visual feedback in the command line immediately after the command was executed. No more than one command can be executed at a time. The system is responding in a sequential manner to the user input.

5. Design

5.1. Class diagram



5.2. Sequence diagrams (for each use case)



5.3. GRASP

What does it mean?

General Responsibility Assignment Software Patterns, abbreviated GRASP, consist of guidelines for assigning responsibility to classes and objects in object-oriented design.

What should I do at this point?

GRASP patterns we identified: Controller, Low Coupling, Polymorphism