Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

15 March, 2019

Developed by:

Name

93X

Version History

Version	Description of Change	Author	Date
V01	Initial Version	Mircea Sorin-Sebastian, Nazarie Ciprian	8 March, 2019

Contents

Analysis and design Document

1. Functional Requirements

List the functional requirements (FR) of the system.

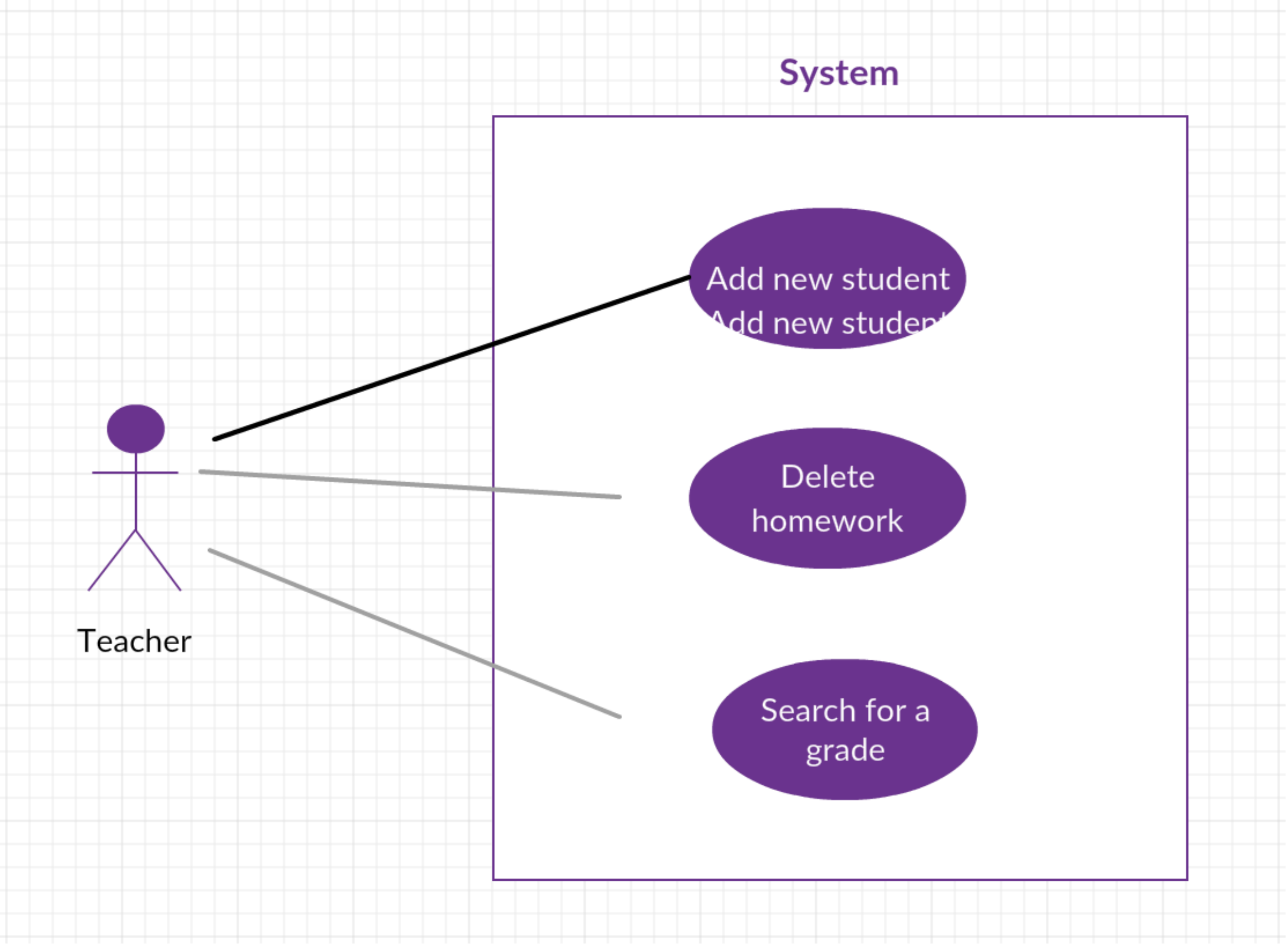
Section/ Requirement ID	Requirement Definition
FR 0	Implement CRUD operations for the Student entity
FR 1	adding a laboratory theme
FR 2	Extending the term of delivery for an existing subject (if the current week number is less than or equal to the number of weeks with the assignment deadline).
FR 3	When adding a new laboratory theme, as well as modifying the delivery date of a theme, all students will be notified by email. The app will offer the ability to unsubscribe from these notifications
FR 4	adding a grade for a particular student to a laboratory topic; any delays due to delays in delivery of a theme will be automatically calculated, showing the student's maximum mark on the topic. Important: A student, on a laboratory theme, has only one grade;
FR 5	When adding a grade, the following information will be retained in the NameStudent.txt file: a "Theme:" ThemeNumber "Delivered in the week:" NumberOfTheDeliveredWeek “Deadline:" NumberOfDeadlineWeek "Feedback:" feedback, suggestions, and explanations in connection with the reduced made regarding the grade.
FR 6	The NameStudent.txt file (or its content) will be emailed to the student, weekly, with the subject "Feedback laboratory MAP".
FR 7	The delays will not be considered if the student has motivation. Also, if the teacher did not enter the notes in time, it will be possible to specify the week in which the subject was delivered.

Section/ Requirement ID	Requirement Definition
FR 8	Filtering entities based on criteria.
FR 9	Reports – Laboratory grade for each student (the weighted average of grades from the lab topics; weight share = number of weeks allocated to the topic). The hardest the theme: the average of the grades on the theme is the smallest. Students who can enter the exam (average greater than or equal to 4). Students who have delivered all the themes on time

2. Actors

The actors are the teachers who manage their classes.

3. Use cases – diagram



1. Use case number 1 (Add new student)

Actors: Teacher

Description: Teacher delete a homework

Precondition: The homework exists (by id)

Postcondition: The homework is deleted

User action	System response
1. Insert 1	
	2. System will show Student Menu
3. Insert 1	
	4. System will require Student Id
5. Inserted student id	
	5. Validates id
	6. System will require Student Name
7. Insert student name	
	8. System will require Group
9. Insert group	
	10. System will require Email
11. Insert Email	
	11. System will give a message if student was successful

Exceptions: When the fields aren’t filled properly.

2. Use case number 2 (Deleting a homework)

Actors: Teacher

Description: Teacher inserts a new student

Precondition: Student doesn’t exist (check by id)

Postcondition: Student will exist

User action	System response
1. Insert 2	
	2. System will show Homework Menu
3. Insert 3	
	4. System will require Homework Id
5. Inserted homework id	
	5. Validates id
	6. System will show the result of the action

3. Use case number 3 (Search for a grade)

Actors: Teacher

Description: Teacher searches for a grade

Precondition: Student doesn’t exist (check by id)

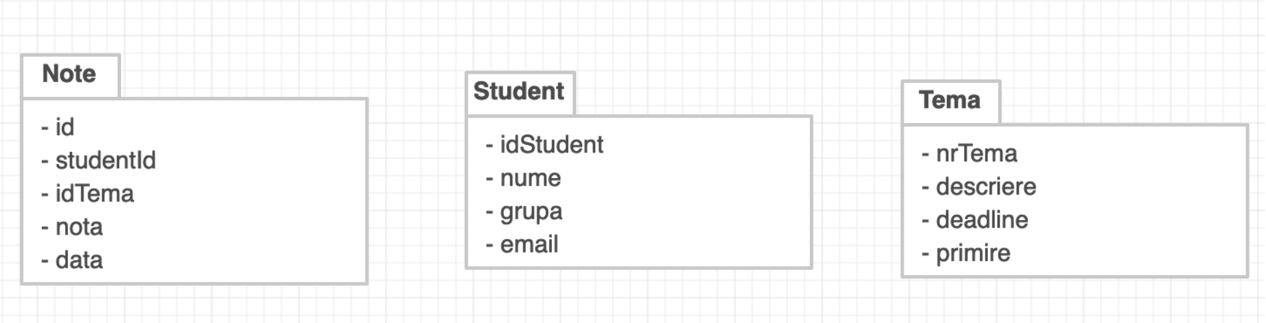
Postcondition: Student will exist

User action	System response
1. Insert 3	
	2. System will show Grades Menu
3. Insert 3	
	4. System will require Student Id
5. Inserted Student id id	
	5. Validates id
	6. System will show the result of the homework number
7. Inserted Homework Number	
	7. System will validate the number

	8. System will show the result of the operation

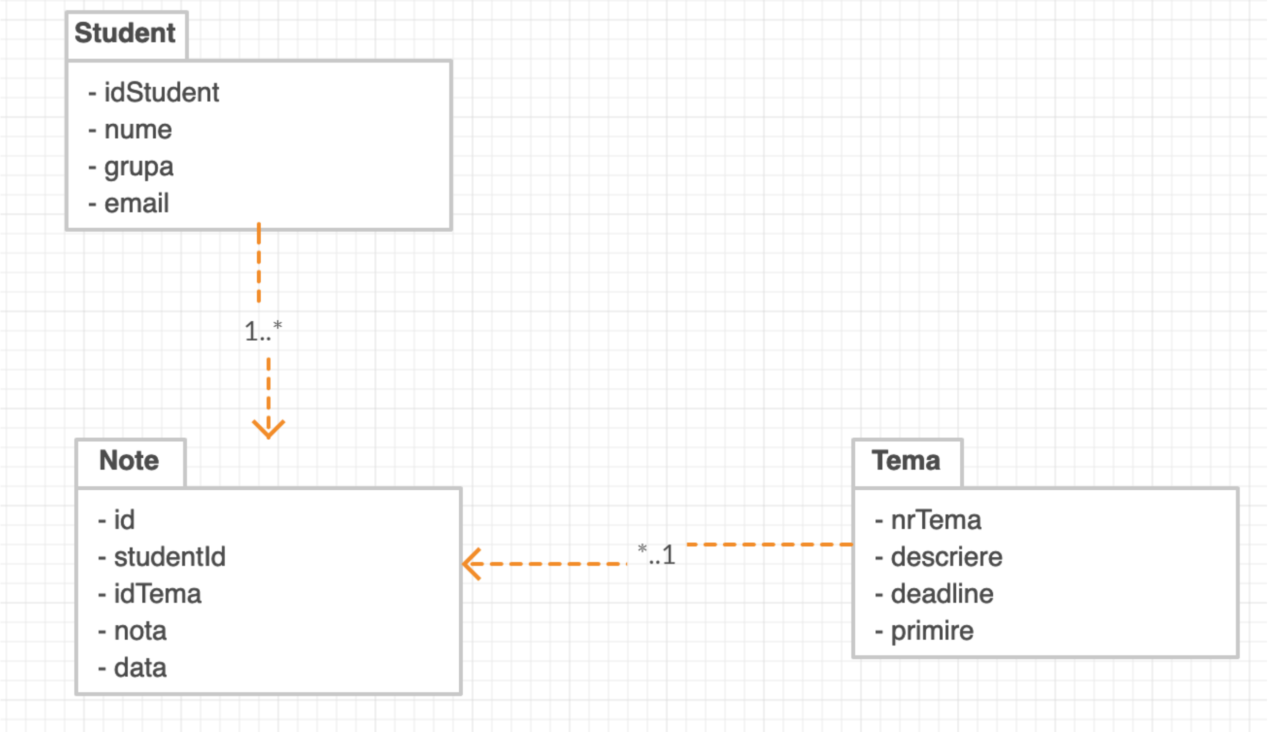
4. Analysis

4.1. Entities



4.2. Relations between entities

Write the relations between the identified entities.



4.3. Attributes

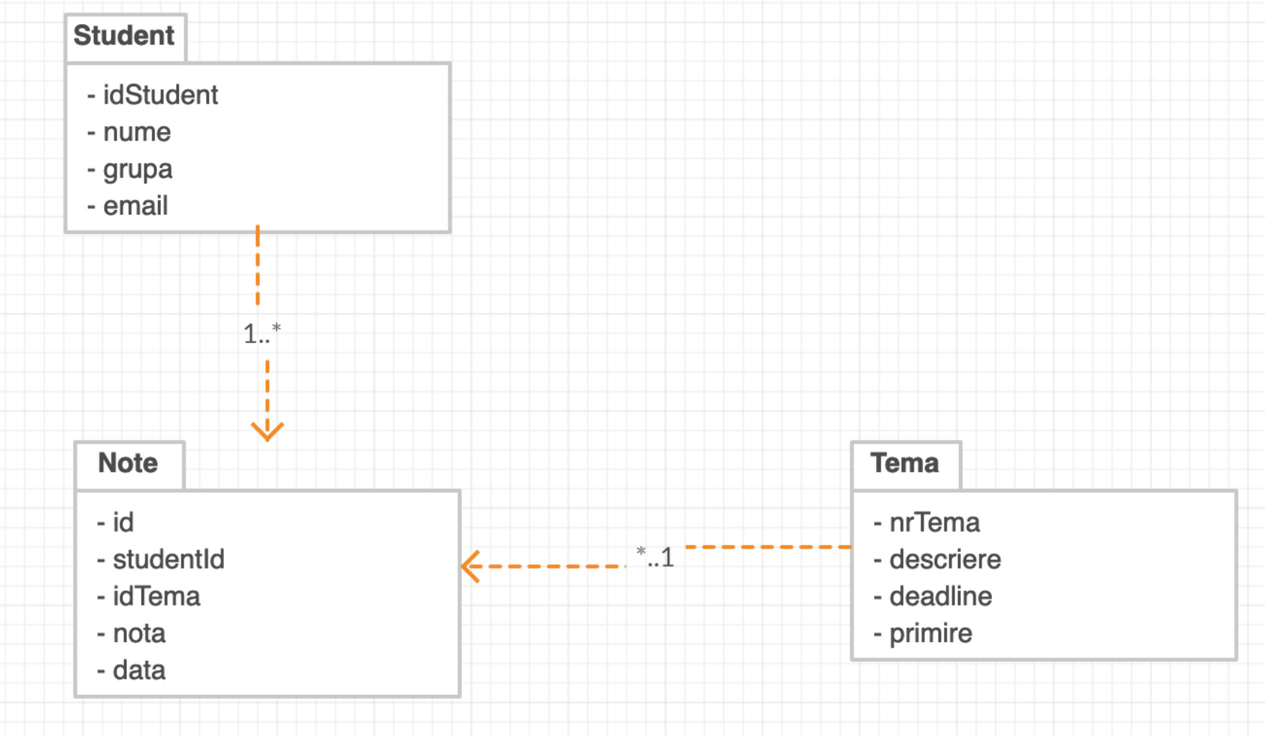
Write the attributes of the identified entities.

Note: idStudent

Students: id, (Foreighn : studentId, idTema)

Homeworks: nrTema

4.4. System behavior



4.4.1. Use case

Use Case 1

Normal behavior: It adds the new student in the lis

Exceptions: Input data is invalid/Student already exists

Use Case 2

Normal behavior: It deletes a homework

Exceptions: Input data is invalid/Homework doesn’t exist

Use Case 3

Normal behavior: It searches for a grate

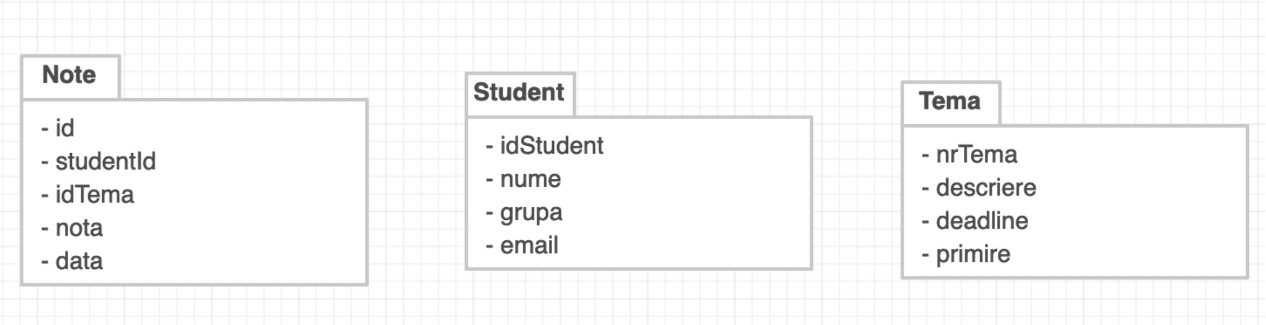
Exceptions: Input data is invalid/Student doesn't exist / Homework number doesn't exist

4.5. System events

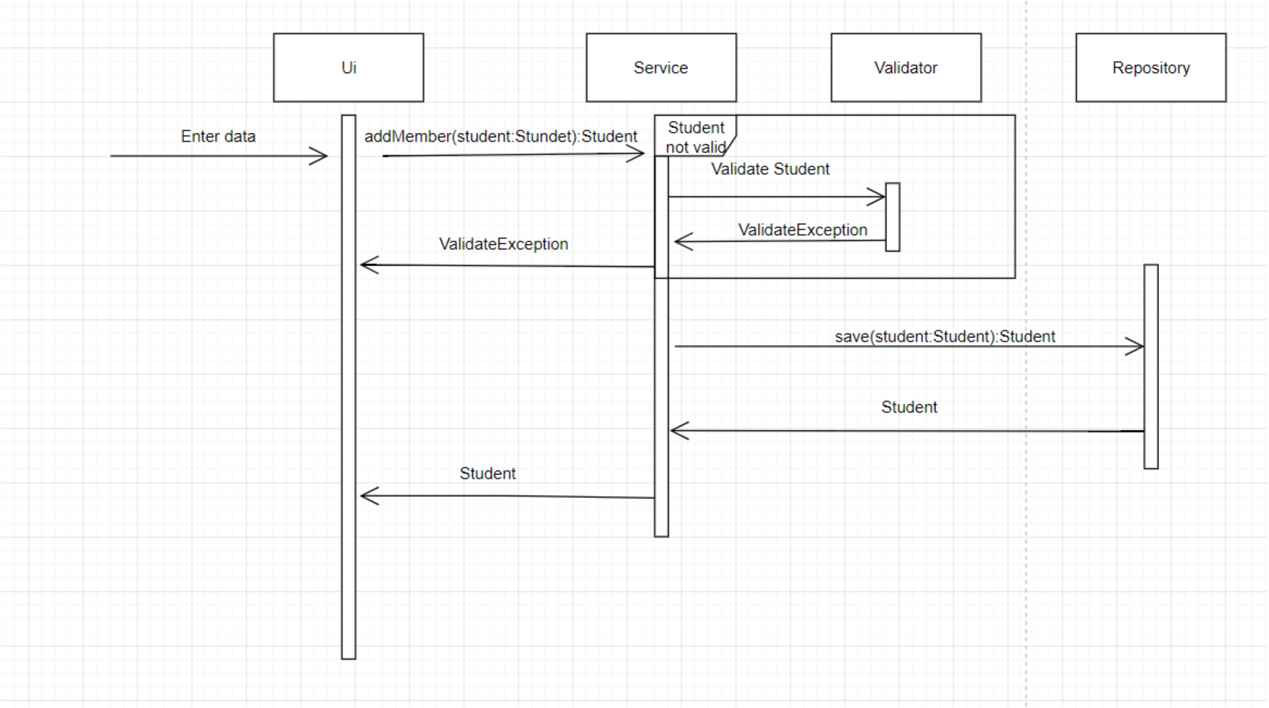
System events are the operations it allows the user to perform (Add new student…), requests for data and notifying the user if the operation was executed successfully.

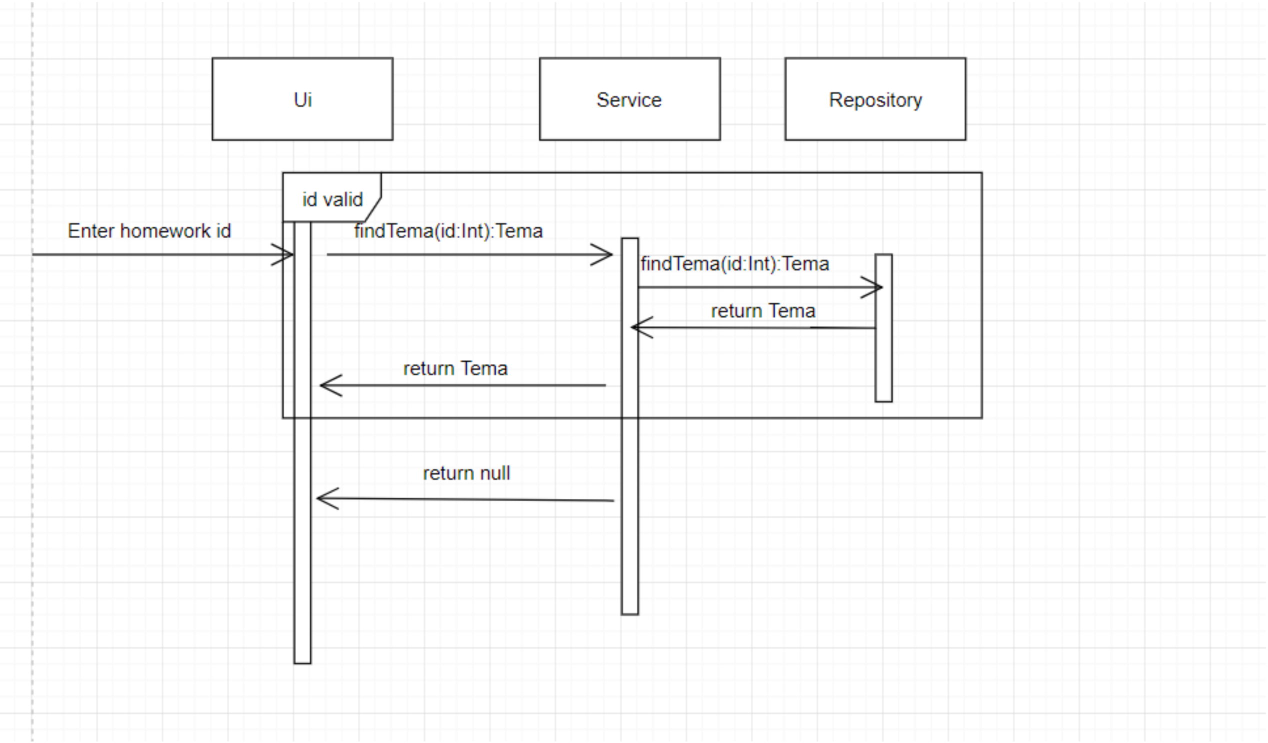
5. Design

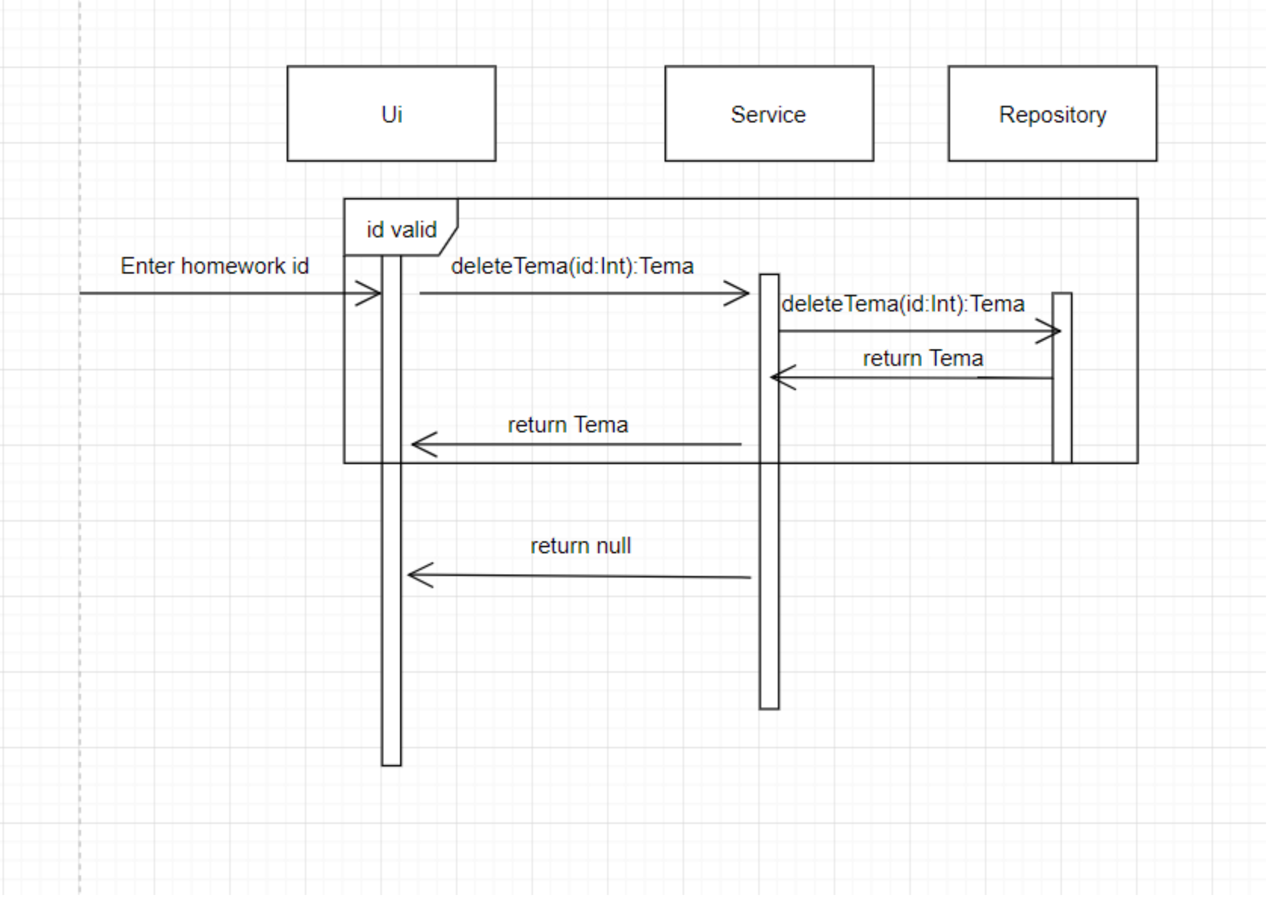
5.1. Class diagram



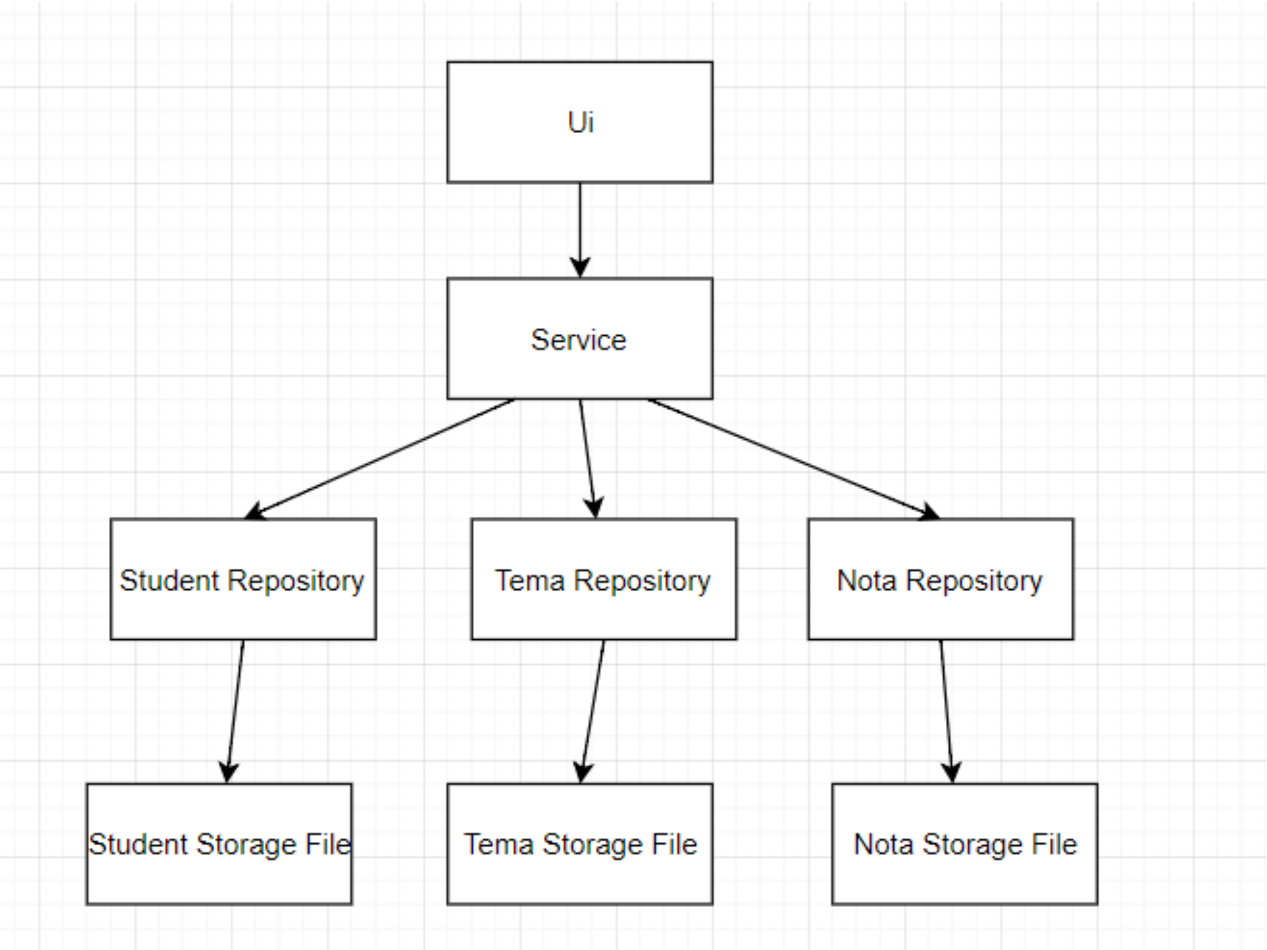
5.2. Sequence diagrams (for each use case)




Ui
Service
Repository
id valid
Enter homework id
findTema(id:Int):Tema
findTema(id:Int):Tema
return Tema
return Tema
return null


Ui
Service
Repository
id valid
Enter homework id
deleteTema(id:Int):Tema
deleteTema(id:Int):Tema
return Tema
return Tema
return null

5.3. GRASP


Ui
Service
Student Repository
Tema Repository
Nota Repository
Student Storage File
Tema Storage File
Nota Storage File