Software Requirements Specification

Exam Grades Management System

Version 1.0 approved

This work was based on the submissions of the project of the course Software Engineering (CS223) and was prepared by

Sohail Khan (B16CS036)

Aniket Parate (B16CS023)

Table of Contents

Revision History

Date	Version	Description	People
18-02-18	1.0	First Draft	Aniket Parate Sohail Khan
11-03-18	2.0	Update draft	Sohail Khan Aniket Parate
24-04-18	3.0	Final Version	Sohail Khan Aniket Parate

1. Introduction

1.1 Purpose

This software package is developed from scratch exclusively for Indian Institute of Technology, Jodhpur in order to

To enable student to view their marks and grades any time from their home. No need to meet the professors personally.

To enable faculty to upload marks and submit grades from his/her office. No need to mail and wait it to be forwarded to all mailing groups.

To manage the complete grading process without any third managing authority.

1.2 Product Scope

The scope of student Grade Management System is

To provide faculty with an easy way to submit grades.

To allow student to quickly access their grades.

This system can be implemented in every institute to smoothen the grades management process.

1.3 Glossary

Student	The user who would be studying in the institute where the system will be implemented.
Faculty	The user who would be teaching any of the courses in the institute where the system would be implemented.

1.4 References

The following references are used in preparing this SRS

The SRS Template and Sample sent by the faculty.
Wikipedia.

2. Overall Description

2.1 Product Functions

This software package is expected to offer the following services:
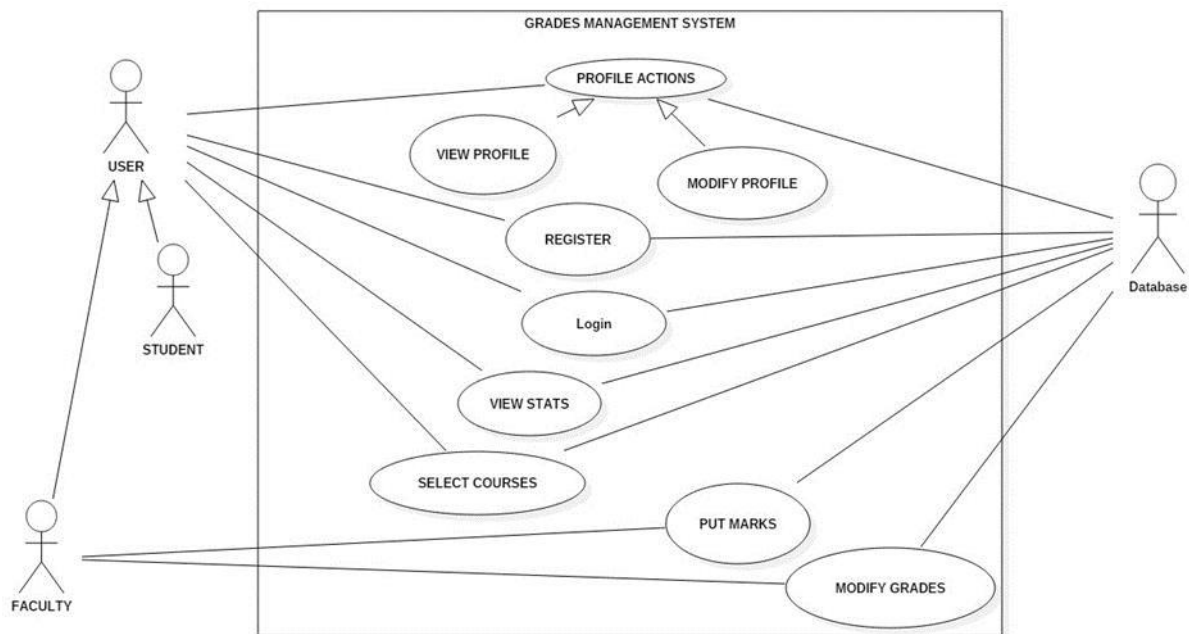
1. For Student:

    a) Facility to view and modify their personal details.
    b) Facility to add their enrolled courses.
    c) Facility to view their marks, grades and pointers.
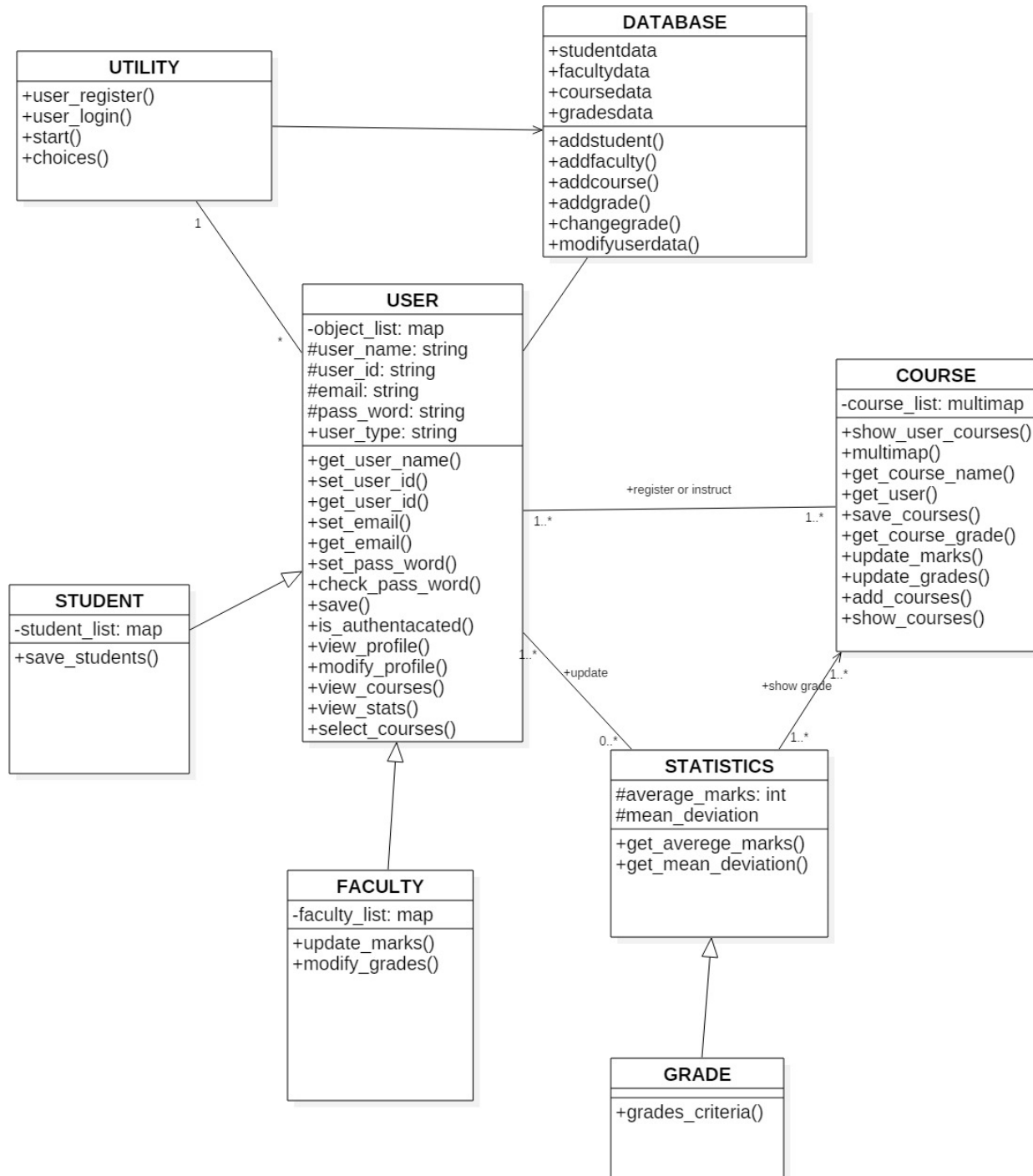
2. For Faculty:

    a) Facility to view and modify their personal details.
    b) Facility to add their teaching courses.
    c) Facility to add student's marks, grades and pointers.

2.2 Use Case Diagram
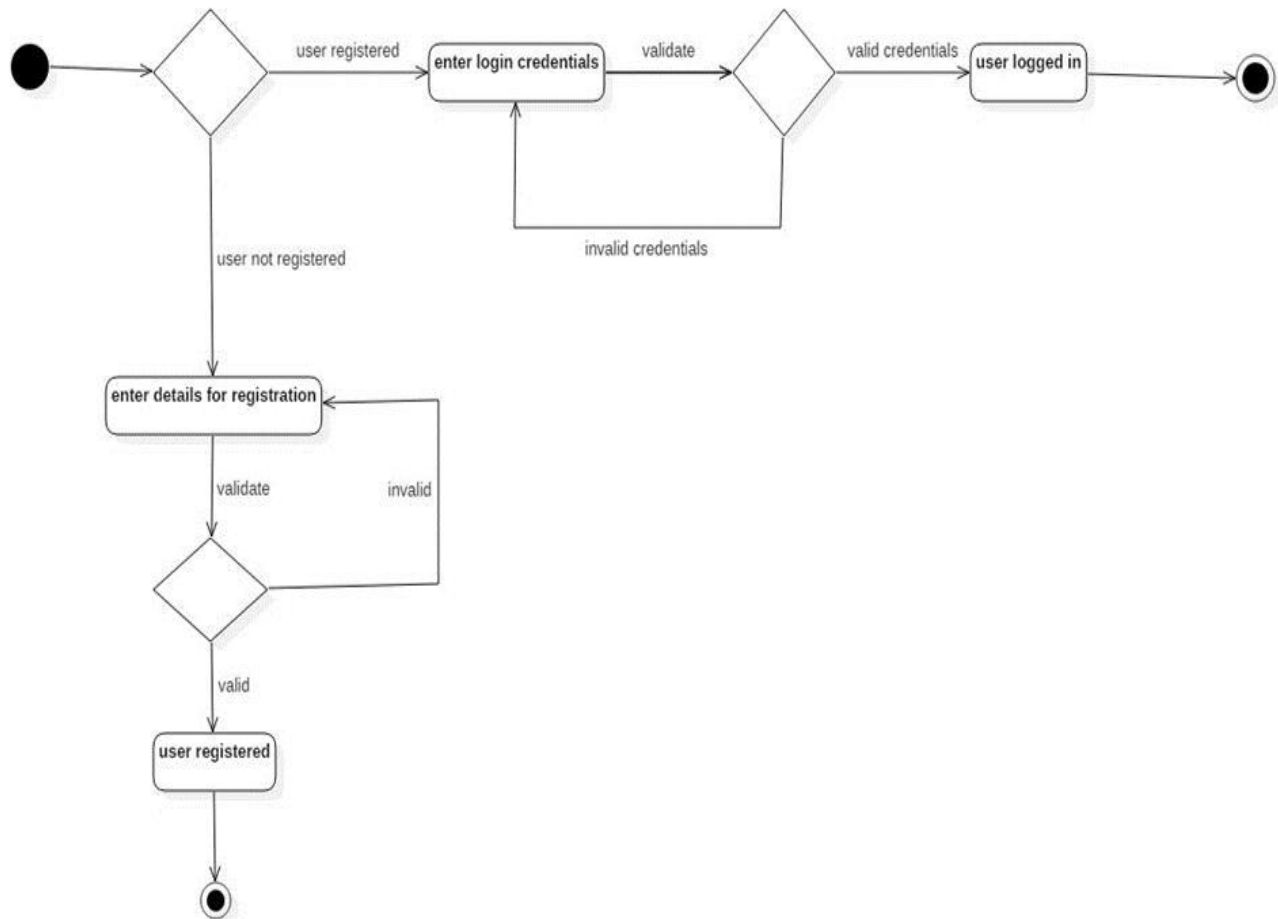
2.3 Class Diagram

UTILITY

+user_register()
+user_login()
+start()
+choices()

DATABASE

+studentdata
+facultydata
+coursedata
+gradesdata

+addstudent()
+addfaculty()
+addcourse()
+addgrade()
+changegrade()
+modifyuserdata()

USER

-object_list: map
#user_name: string
#user_id: string
#email: string
#pass_word: string
+user_type: string

+get_user_name()
+set_user_id()
+get_user_id()
+set_email()
+get_email()
+set_pass_word()
+check_pass_word()
+save()
+is_authentacated()
+view_profile()
+modify_profile()
+view_courses()
+view_stats()
+select_courses()

COURSE

-course_list: multimap

+show_user_courses()
+multimap()
+get_course_name()
+get_user()
+save_courses()
+get_course_grade()
+update_marks()
+update_grades()
+add_courses()
+show_courses()

STUDENT

-student_list: map

+save_students()

FACULTY

-faculty_list: map

+update_marks()
+modify_grades()

STATISTICS

#average_marks: int
#mean_deviation

+get_averege_marks()
+get_mean_deviation()

GRADE

+grades_criteria()

+register or instruct

1..*   1..*
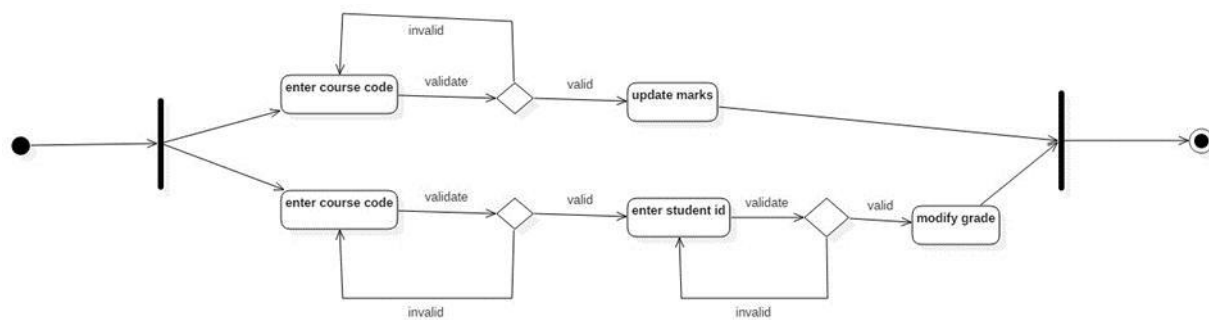
1

*

1..*

+update

1..*

0..*

+show grade

1..*

1..*

2.4 Activity Diagram
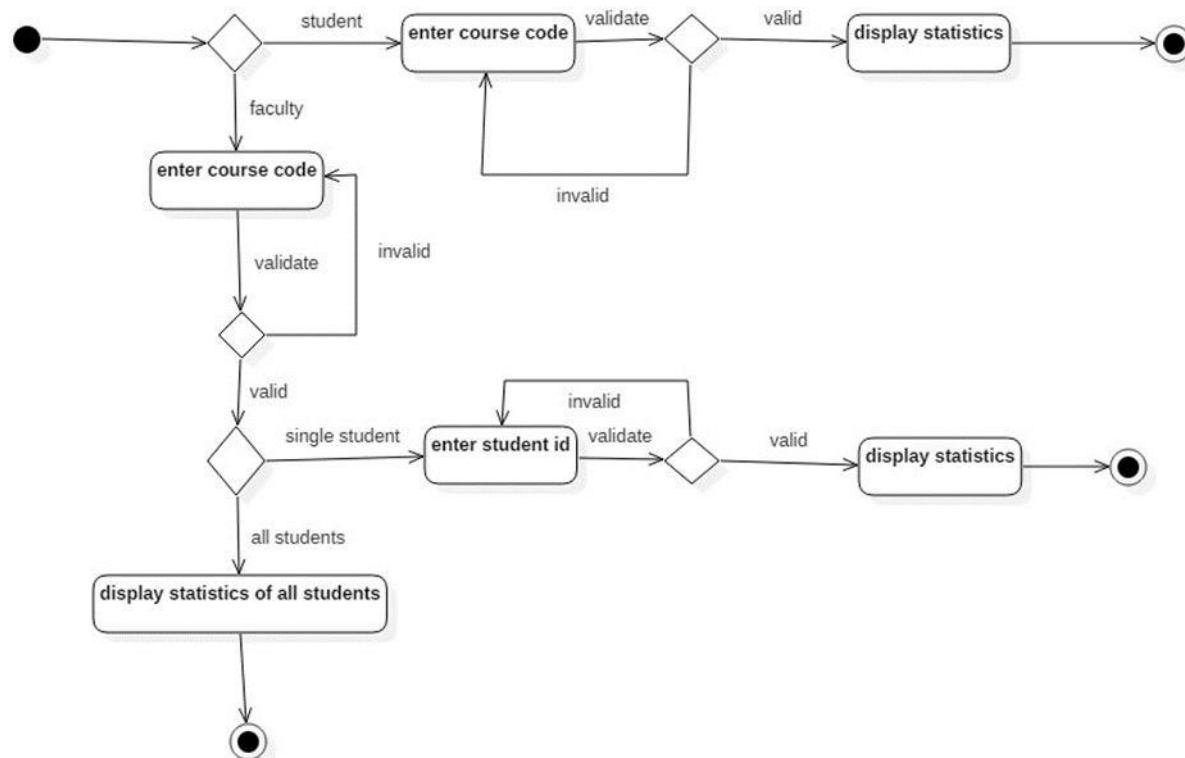
1. Activity diagram for login/register activity

2.Activty diagram for update marks / modify grades
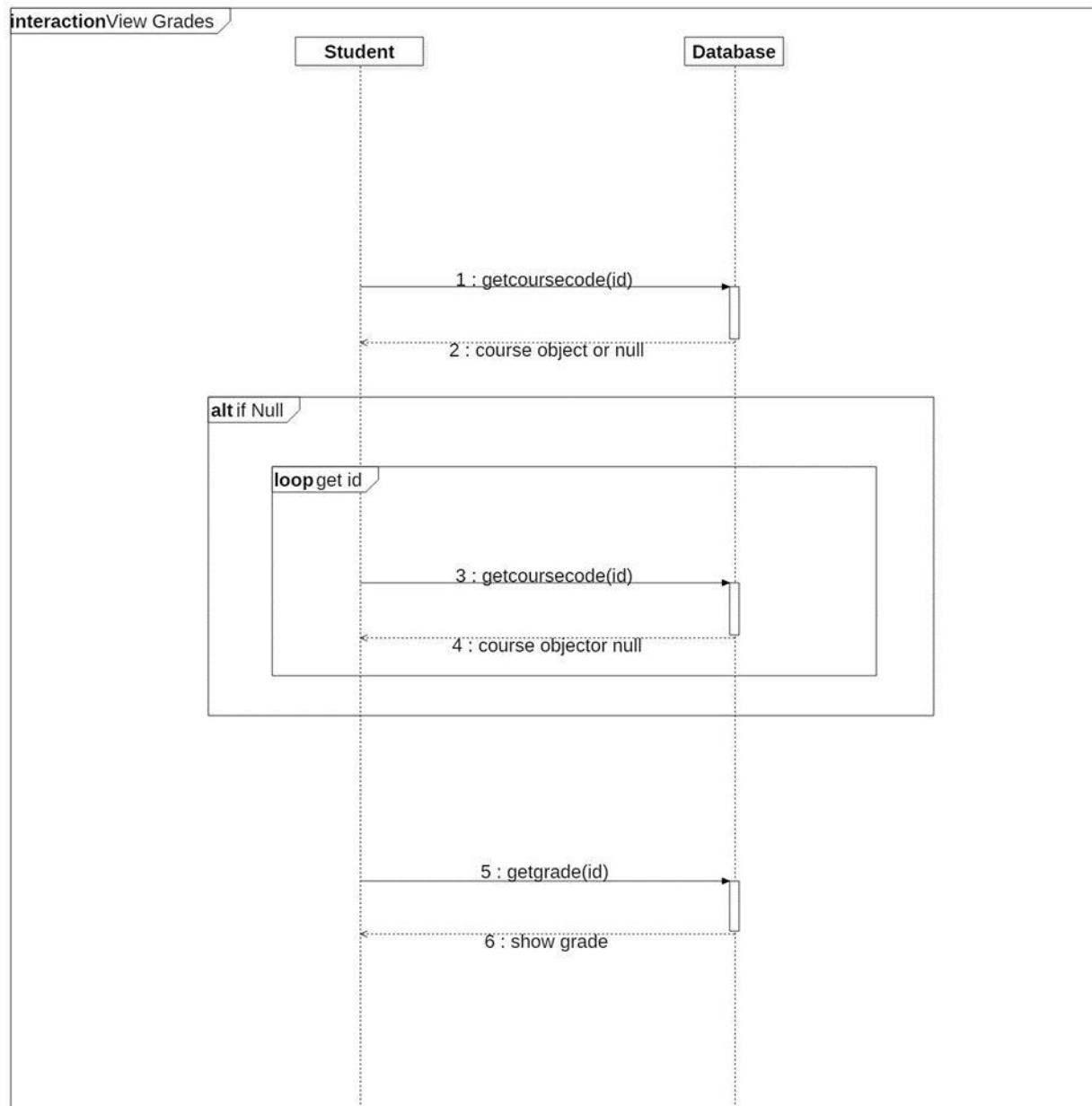   Precondition is that the user is registered and is faculty.

3.Activity diagram for view statistics activity after the registered user is logged in.

2.5 Sequence Diagram

1. Sequence Diagram for view Grades.

2. Sequence diagram for Grades update

interaction Add marks

| Faculty | Database |

1 : getcoursecode(id)

2 : course object or null

alt if Null

loop get id

3 : getcoursecode(id)

4 : course objector null

loop get student id

5 : getstudentid(id)

6 : student object

7 : updatemarks(id)

8 : marks updated
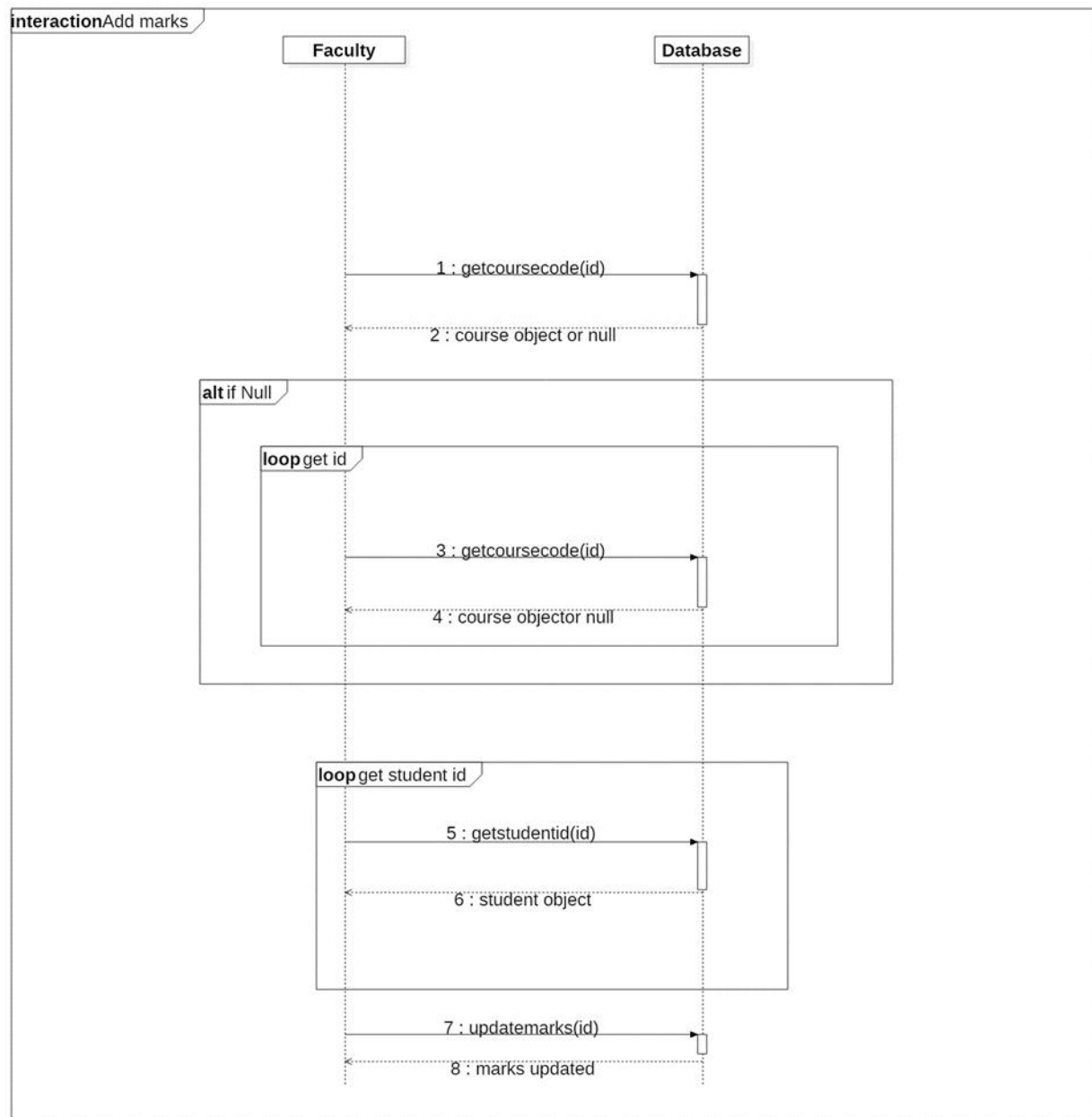
2.6 Assumptions and Dependencies

1. The list of courses to be selected by both faculty and students is provided at the start of the semester.

2. Grades should be given to each students registered in a particular course.

3. The following are the valid grades: A,B,C,D,F.

4. Grades are calculated by the following formula A-

   Marks > [average marks + 1.5*(mean deviation)]

   B- Marks between [average marks + mean deviation] and [average marks + 1.5*(mean deviation)]

   C- Marks between [average marks] and [average marks + mean deviation]

   D- Marks between [average marks – mean deviation] and [average marks]

   E- Invalid grade

   F- Marks less than [average marks – mean deviation]

3. System Requirements

The requirements for the student Grade Management system are as following:

3.1 Use Case Description

3.1.1 Login

Use Case Name	Login
Actors	Student, Faculty, Database
Trigger	User fills the login credentials.
Precondition	The User must be registered.
Basic Path	User enters its user ID and Password.
Alternative Path	No alternative path.
Post Condition	The user is taken to his profile page.
Exception Paths	All fields must be filled.
Others	None

3.1.2 Register

Use Case Name	Register
Actors	Student, Faculty, Database
Trigger	User chooses register option.
Precondition	The User must have an institute LDAP credential.
Basic Path	User enters his details.
Alternative Path	No alternative path.
Post Condition	The user is registered successfully.
Exception Paths	All fields must be filled.
Others	None

3.1.3 Modify Profile

Use Case Name	Modify Profile
Actors	Student, Faculty, Database
Trigger	User chooses modify profile option from his profile action menu.
Precondition	The user must be logged in.
Basic Path	User enters his updated personal details.
Alternative Path	No alternative path.
Post Condition	The user is taken (back) to his profile page.
Exception Paths	All fields must be filled.
Others	None

3.1.4 View Profile

Use Case Name	View Profile
Actors	Student, Faculty, Database
Trigger	User chooses view profile option from his Profile action menu.
Precondition	The user must be logged in.
Basic Path	Database retrieves user information.
Alternative Path	No alternative path.
Post Condition	The user is taken to his profile page.
Exception Paths	None.
Others	None

3.1.5 Select Courses

Use Case Name	Select Courses
Actors	Student, Faculty
Trigger	User chooses select courses option.
Precondition	The user must be logged in.
Basic Path	User chooses enrolled/teaching courses from the list of courses.
Alternative Path	No alternative path.
Post Condition	User gets the list of enrolled/teaching courses.
Exception Paths	None
Others	None

3.1.6 Put Marks

Use Case Name	Put Marks
Actor	Faculty, Database
Trigger	User chooses put marks option.
Precondition	The user must be logged in.
Basic Path	User selects the course from the list of teaching courses and then enters marks in the database.
Alternative Path	No alternative path.
Post Condition	Database stores the marks of selected courses.
Exception Paths	All fields must be filled.
Others	None

3.1.7 Modify Grades

Use Case Name	Add Grades
Actor	Faculty, Database
Trigger	User chooses modify grades option.
Precondition	1. The user must be logged in. 2. The user must put marks.
Basic Path	User select the course from the list of teaching courses and then enters grades for all students(by id).
Alternative Path	No alternative path.
Post Condition	Database stores the grades of selected courses.
Exception Paths	None
Others	None

3.1.8 View Stats

Use Case Name	View Stats
Actor	Student, Professor, Database
Trigger	User chooses view stats option.
Precondition	1. The user must be logged in. 2. Faculty must put marks.
Basic Path	User select course from the list of selected courses and Database retrieves stats of the courses selected.
Alternative Path	No alternative path.
Post Condition	User gets the stats of the selected courses.
Exception Paths	None
Others	None

4. Non-functional Requirements:

1. The system is supposed to provide a platform for grades management thus it shall be scalable to the institute level.

2. Sufficient network bandwidth is required for all the functions to work smoothly.

3. System should be efficient enough to handle all the students at the same time.