Software Requirements Specification

Version 1.2

July 27, 2022

Objective Examination Management System

Submitted in partial fulfillment
Of the requirements of
IT-3003 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223).

The students who submitted this team projects were

- **Sandeep Kumar Swain (20051025)**
- **Shivam Mishra (20051028)**
- **Shivansh Maheswari (20051029)**
- **Shubhangi Singh (20051034)**
- **Ujjawal Singh (20051045)**

Table of Contents

Table of Contents
List of Figures
1.0. Introduction
1.1. Purpose
1.2. Scope of Project
1.3. Constraints
1.4. Assumptions and Dependencies
1.5. Glossary
1.6. References
1.7. Overview of Document
2.1 System Environment
2.2 Functional Requirements Specification
2.3 User Characteristics
2.4 Non-Functional Requirements
3.0. Requirements Specification
3.1 Functional Requirements
3.1.1 Add Student
3.1.2 Remove Student
3.1.3 Edit Student
3.1.4 Add Faculty
3.1.5 Remove Faculty
3.1.6 Edit Faculty
3.1.7 Add Course
3.1.8 Remove Course
3.1.9 Edit Course
3.1.10 Enroll Student in Course
3.1.11 Add faculty to course
3.1.12 Display Profile
3.1.13 Display Courses
3.1.14 Create new exam
3.1.15 View performance of the class
3.1.16 View profile of individual student
3.1.17 Display Student Profile
3.1.18 Display Courses
3.1.19 View Performance
3.1.20 Take New Exam
3.1.21 Login
3.2 Detailed Non-Functional Requirements
3.4 *Logical Structure of the Data*

Revision History

Date	Reason For Changes	Version
28 Jan, 2016	Added UML class, sequence, activity and use case diagrams. Updated Assumptions and Dependencies	1.1
22 Mar, 2016	Added logical structuring of data	1.2

1.0. Introduction

1.1. Purpose

The purpose of this document is to explain various functionalities of the Online Objective Examination System. It provides a detailed overview of the system and explains its uses from the perspective of various types of users involved. It is written for the use of clients, developers and all other parties involved in the development process.

1.2. Scope of Project

This software system will be an Online Objective Examination System designed for use in schools and colleges. This system will be developed in order to minimize the lengthy formalities involved in question paper creation, candidate registration process, evaluating responses and publishing the results. It will be designed in order to reduce the time, cost and work load of staff involved in examination procedure. It reduces the usage of paper and is hence eco-friendly.

The aim of the software system is to automate the Objective Examination system. It would ensure smooth interaction of three types of users involved, namely student, faculty and administrator. The students would be able take the examination and receive a detailed performance report. Faculty would be able to create new question papers. Admin user will have access to profiles of all student and faculty users, right to add or remove users, add or drop students to enrolled courses, assign courses to faculty, add or remove courses and faculty.

1.3. Constraints

All users must compulsorily use their LDAP login (the database will be merged with the one of student registration) details to log in to the Online Objective Examination System. The number of users that can be handled is limited by the software used in development and the capacity of the server on which the software system is hosted. The system is for use only in a particular institute and is limited by the number of computers in the computer center. It also requires constant internet access.

1.4. Assumptions and Dependencies

The system assumes that the registration of faculty, students and courses has already been done by some other system which will not be created as a part of this system. Hence it assumes the information to be directly available to it. The staff of an institute is equipped with a server either at the site or offsite. They will input all the information at the time of installation like information of students, faculties and courses. The institute has a Computer Centre where students will arrive to give the tests. The computers should be equipped with modern browsers. The paper is supposed to be only objective and all questions are of 1 mark each. The faculty will not have an option to add any subjective question to the system. The computer center has the capacity to accommodate all the students of a course.

1.5. Glossary

Term	Definition
User	An admin, student or faculty using the system.

Admin	The system administrator- person with complete rights to alter the details of anyone using the system and add new users as well.
Database	The organized collection of all details in the system.
Credentials	Details used by a user to gain access to the system.

1.6. References

IEEE. *IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.7. Overview of Document

Chapter 2 gives an overall description of the web application followed by the system environment, functional and non-functional requirements and various use cases.

Chapter 3 explains in detail the various functional and non-functional requirements feature wise, including the logical structure of data.

2.0. Overall Description

2.1 System Environment

The System basically interacts with 3 types of people:

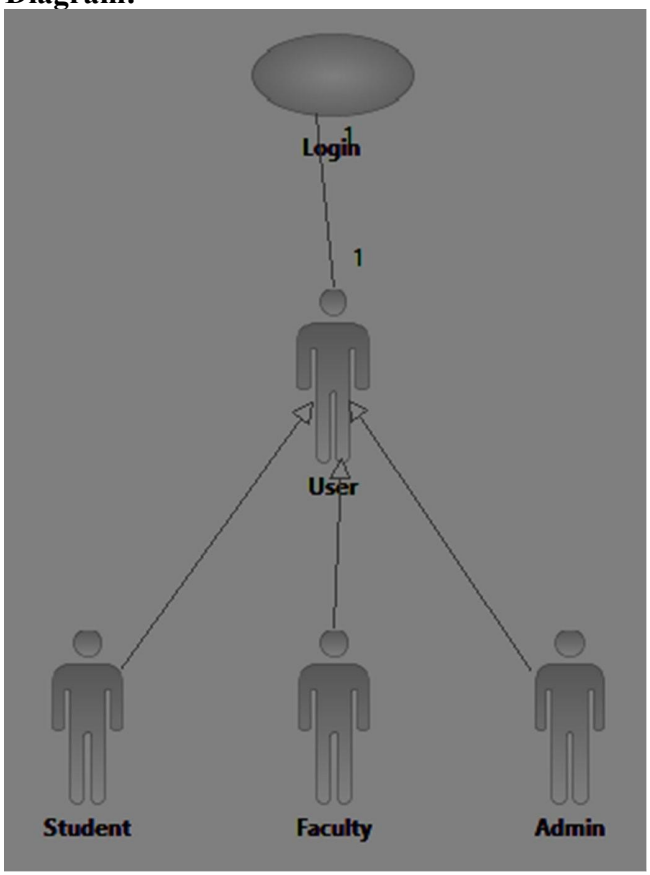
a) Students, who is the user of the software and will give an online test.
b) Faculty, who is the user of the software and will prepare an online test for the student user.
c) Admin, who is responsible for maintaining the database as he should perform various tasks like editing, adding, deleting a user.

2.2 Functional Requirements Specification

The system is basically meant to conduct an online examination for students studying in an institute. They can give one time test for the course they are registered to. Also there has to be an option for preparing the exam also for the students. A faculty can create exams for the courses taught by him/her.

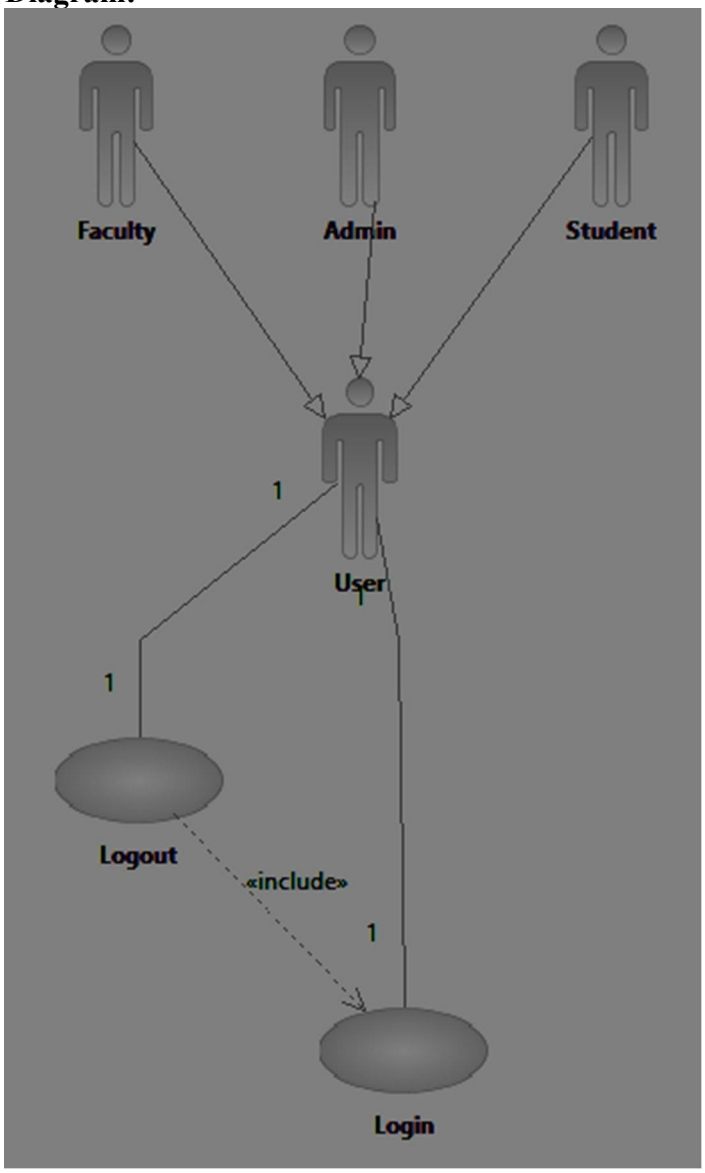
2.2.1 **Use case:** Login

Diagram:



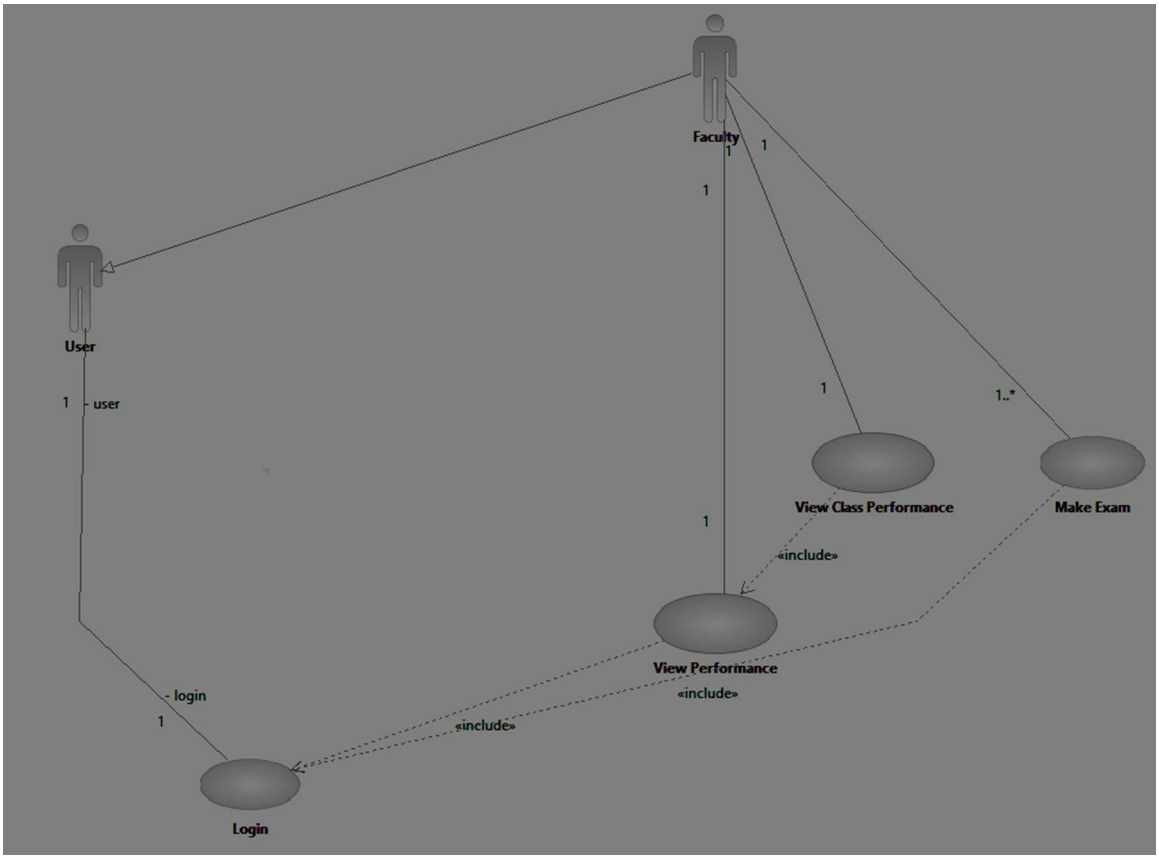
2.2.2 **Use case:** Logout

Diagram:



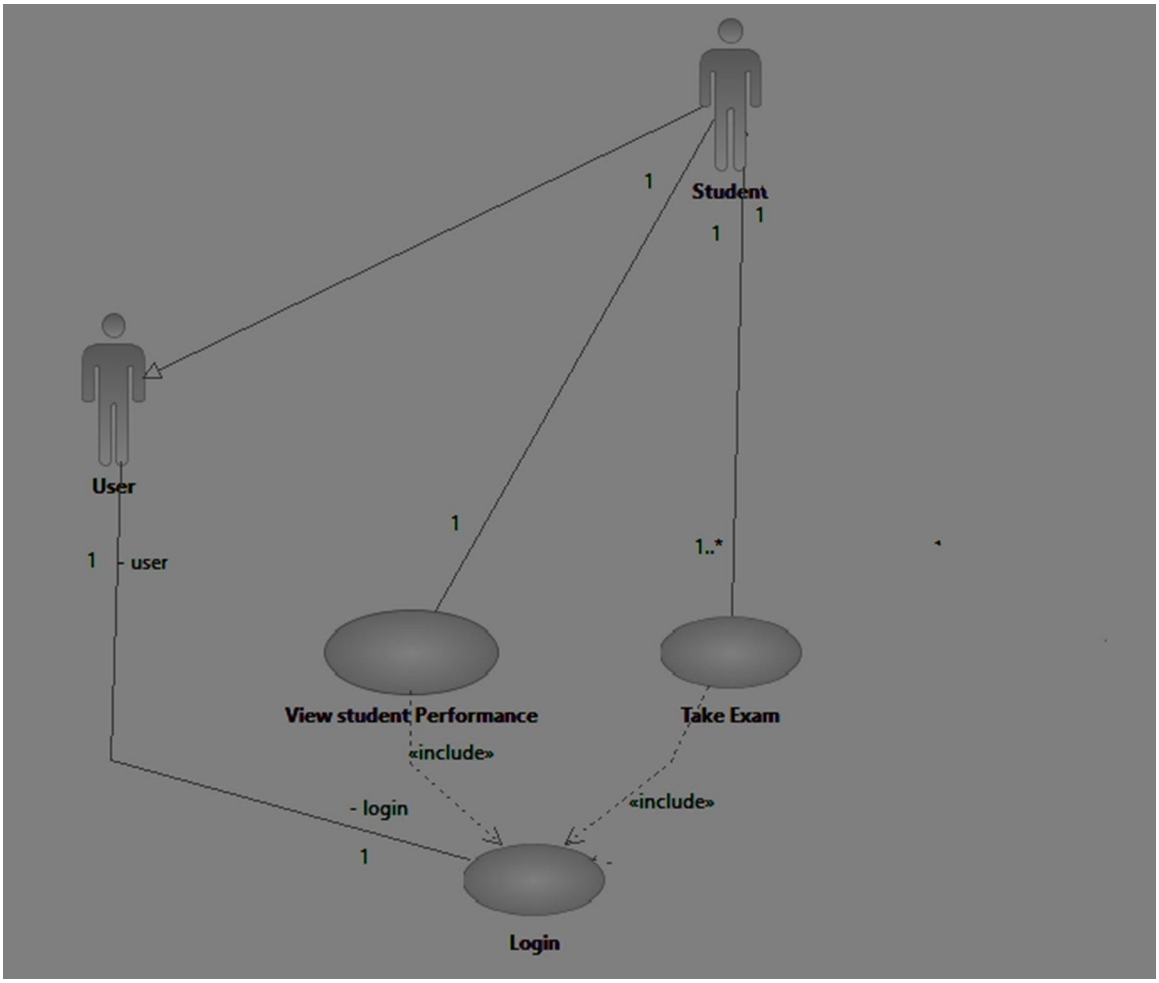
2.2.3 **Use case:** Faculty Use Case

Diagram:



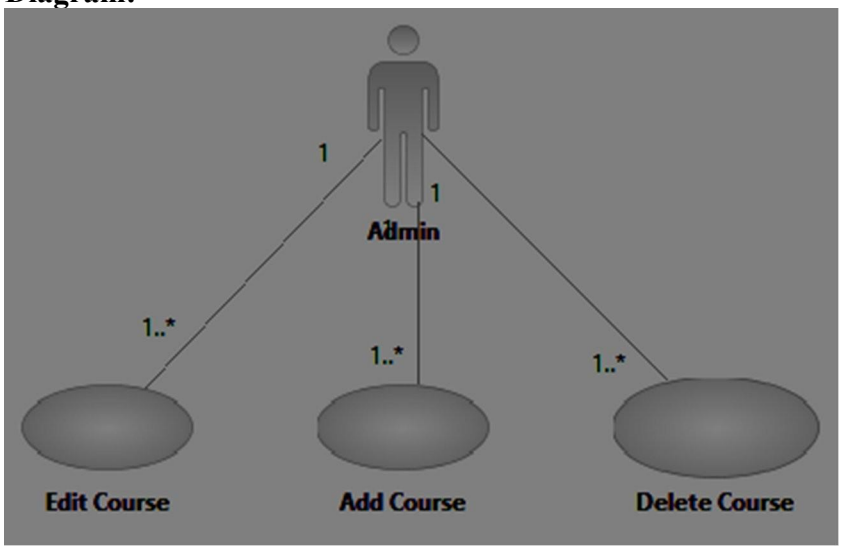
2.2.4 **Use case:** Student Use Case

Diagram:



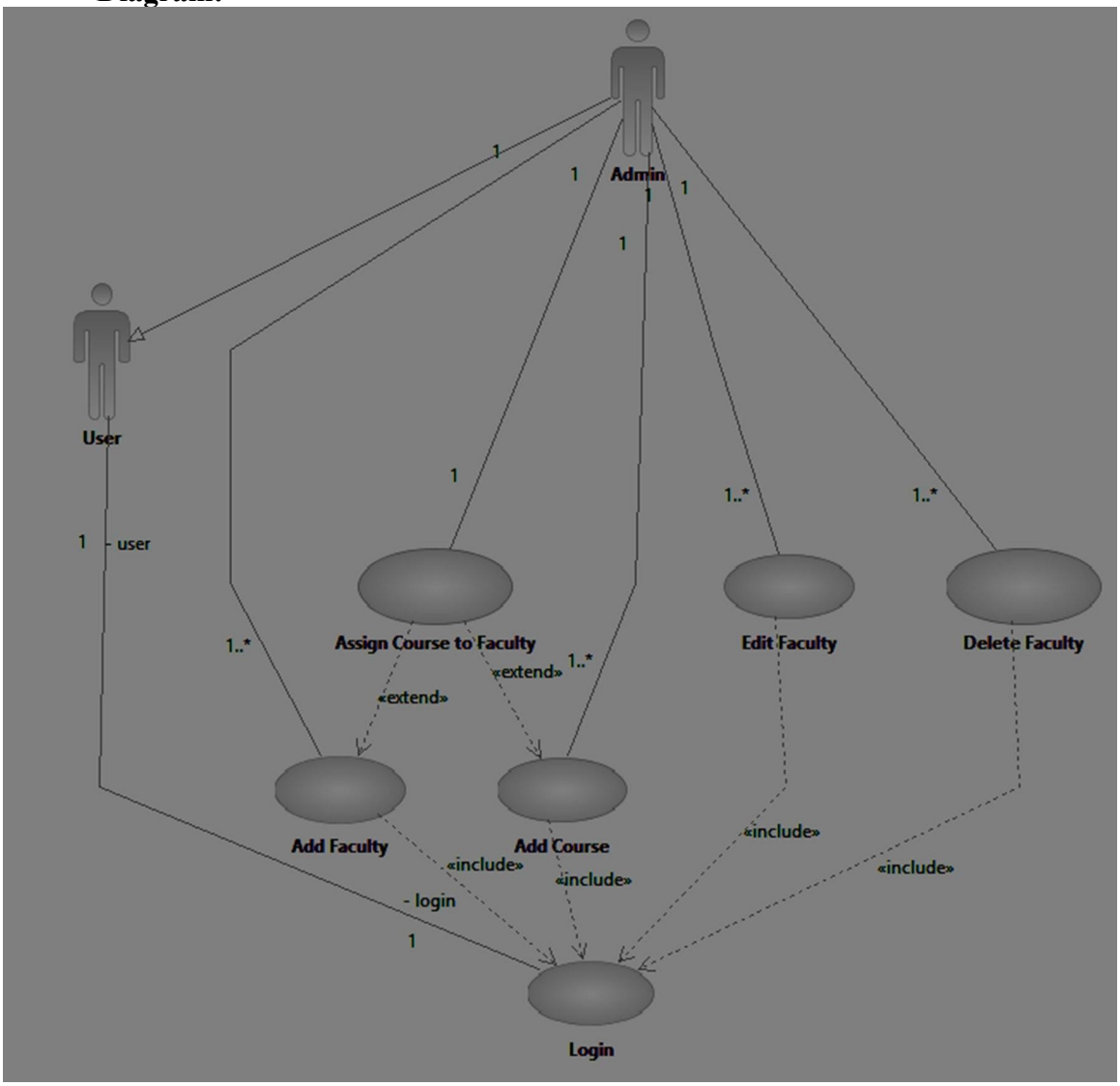
2.2.5 **Use case:** Manage Course

Diagram:



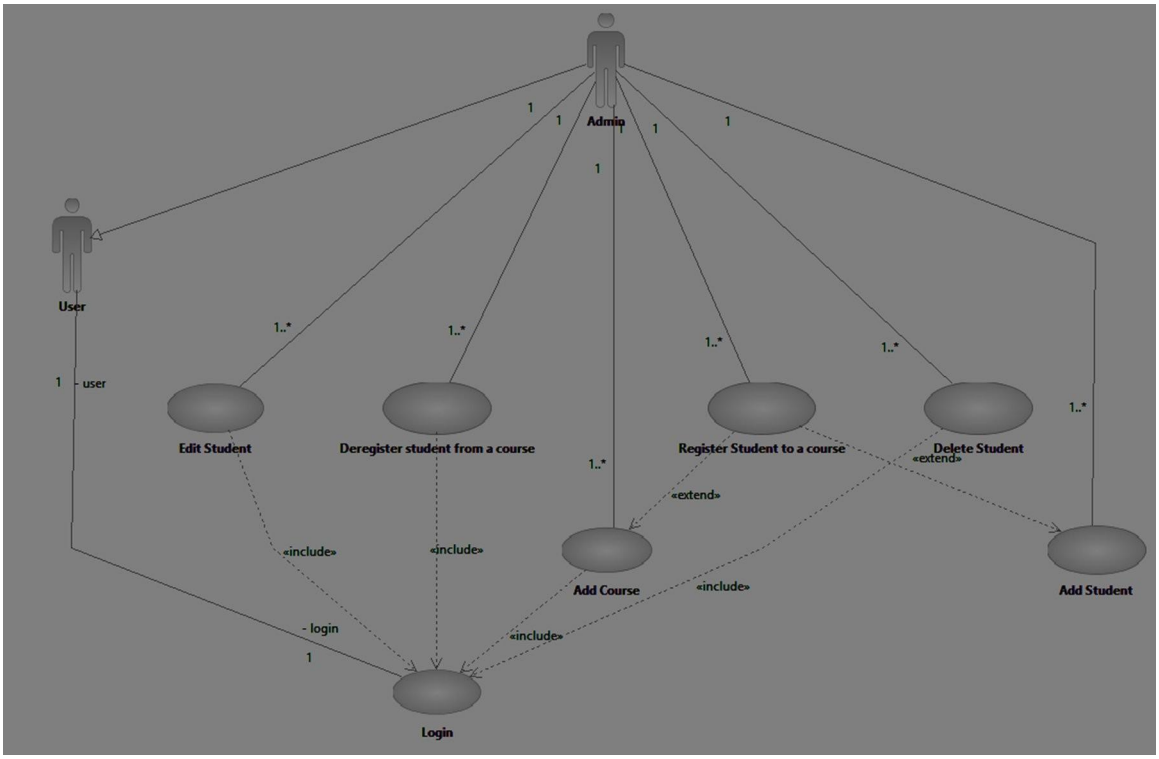
2.2.6 **Use case:** Manage Faculty

Diagram:

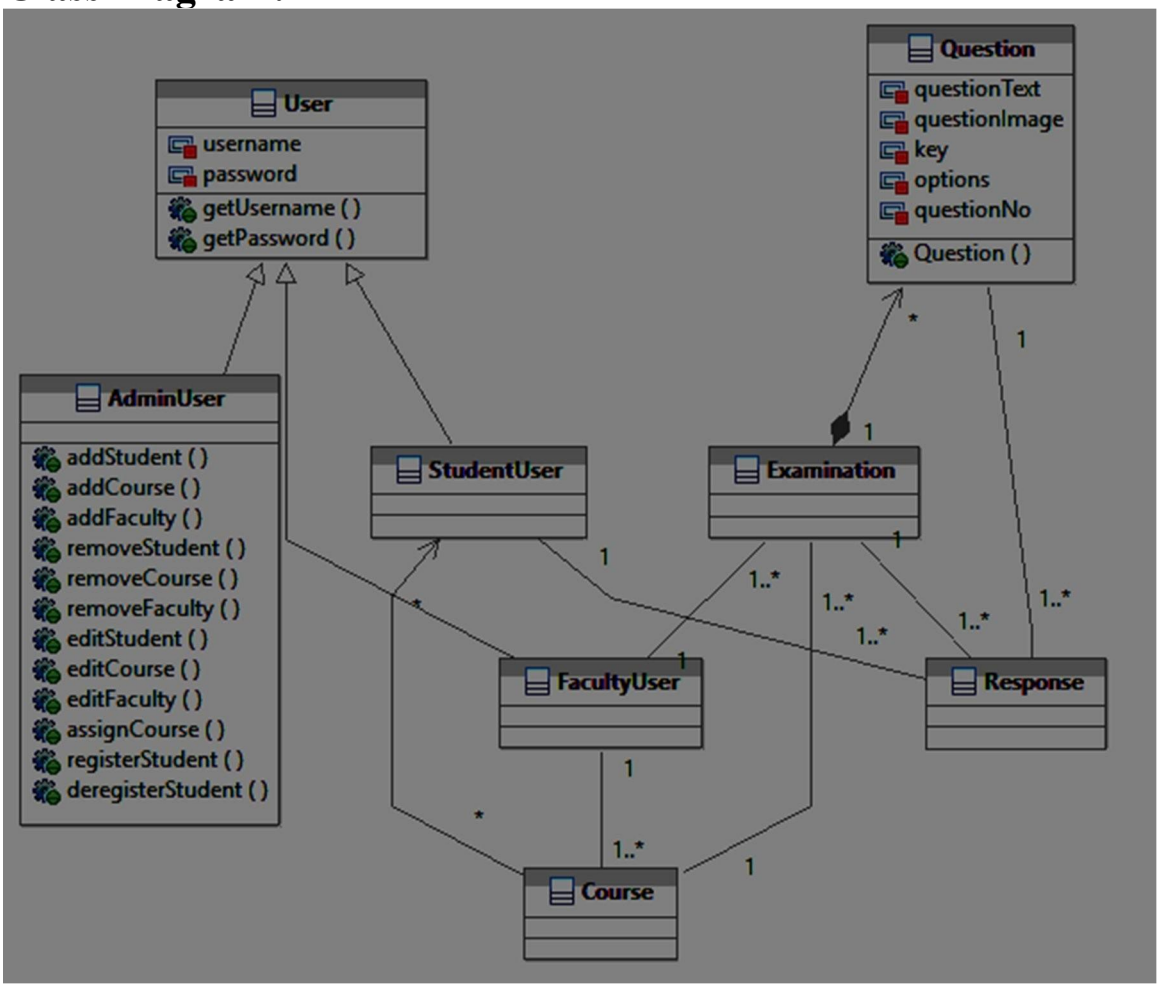


2.2.7 **Use case:** Manage Student

Diagram:

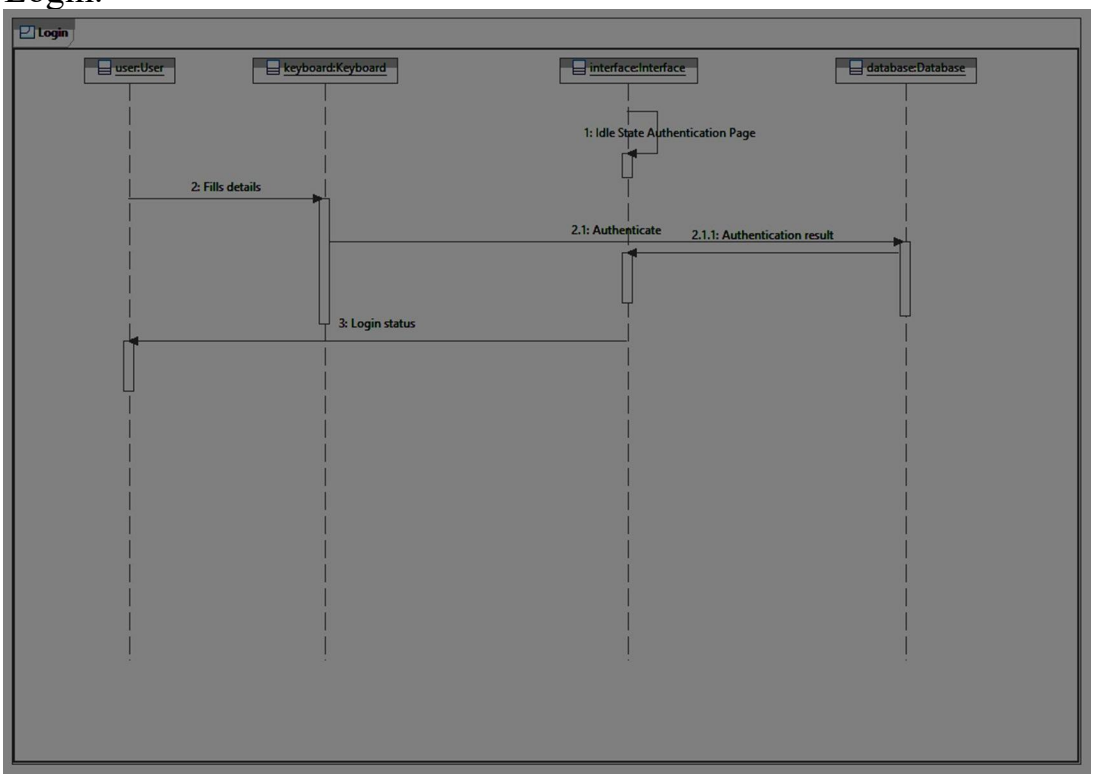


Class Diagram:

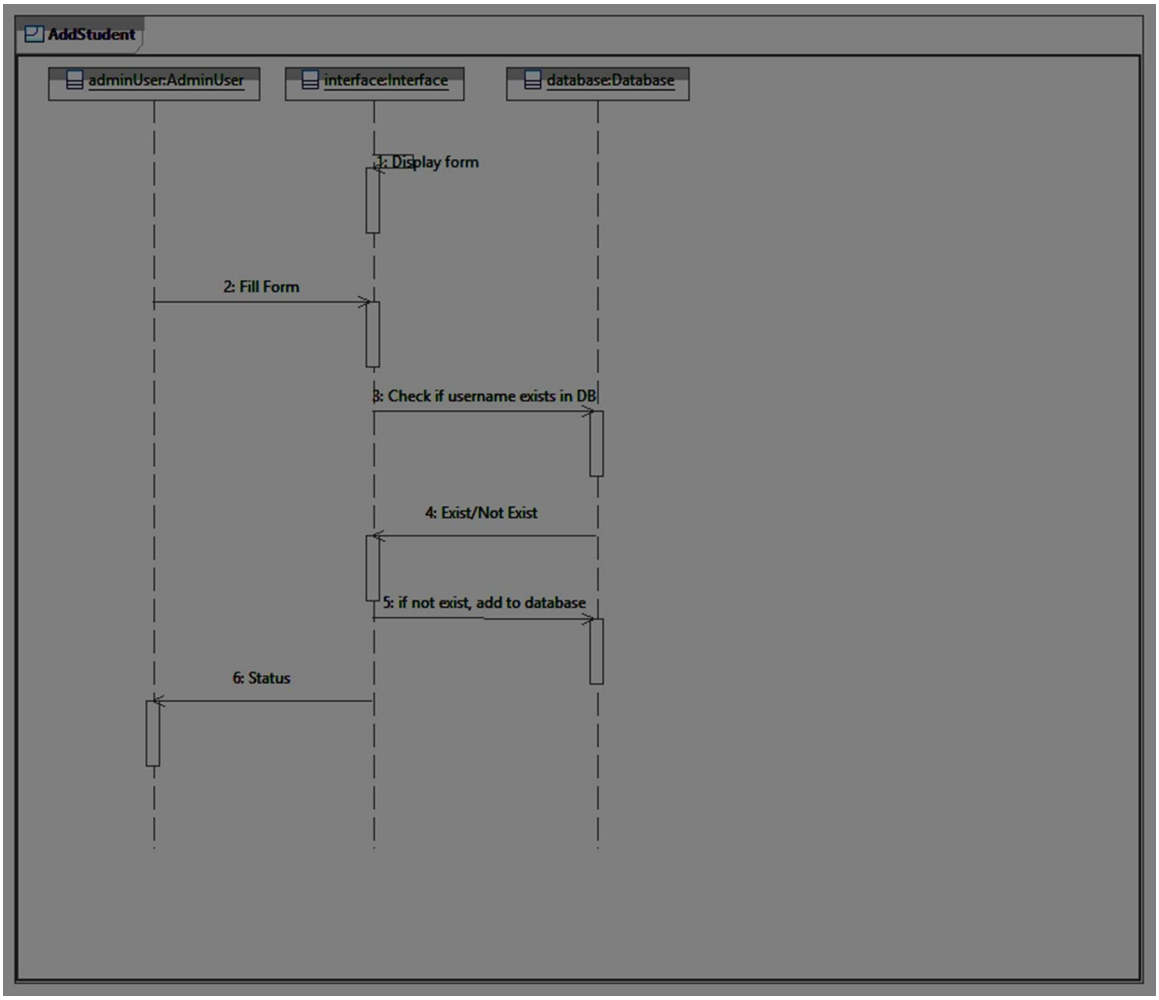


Sequence Diagrams:

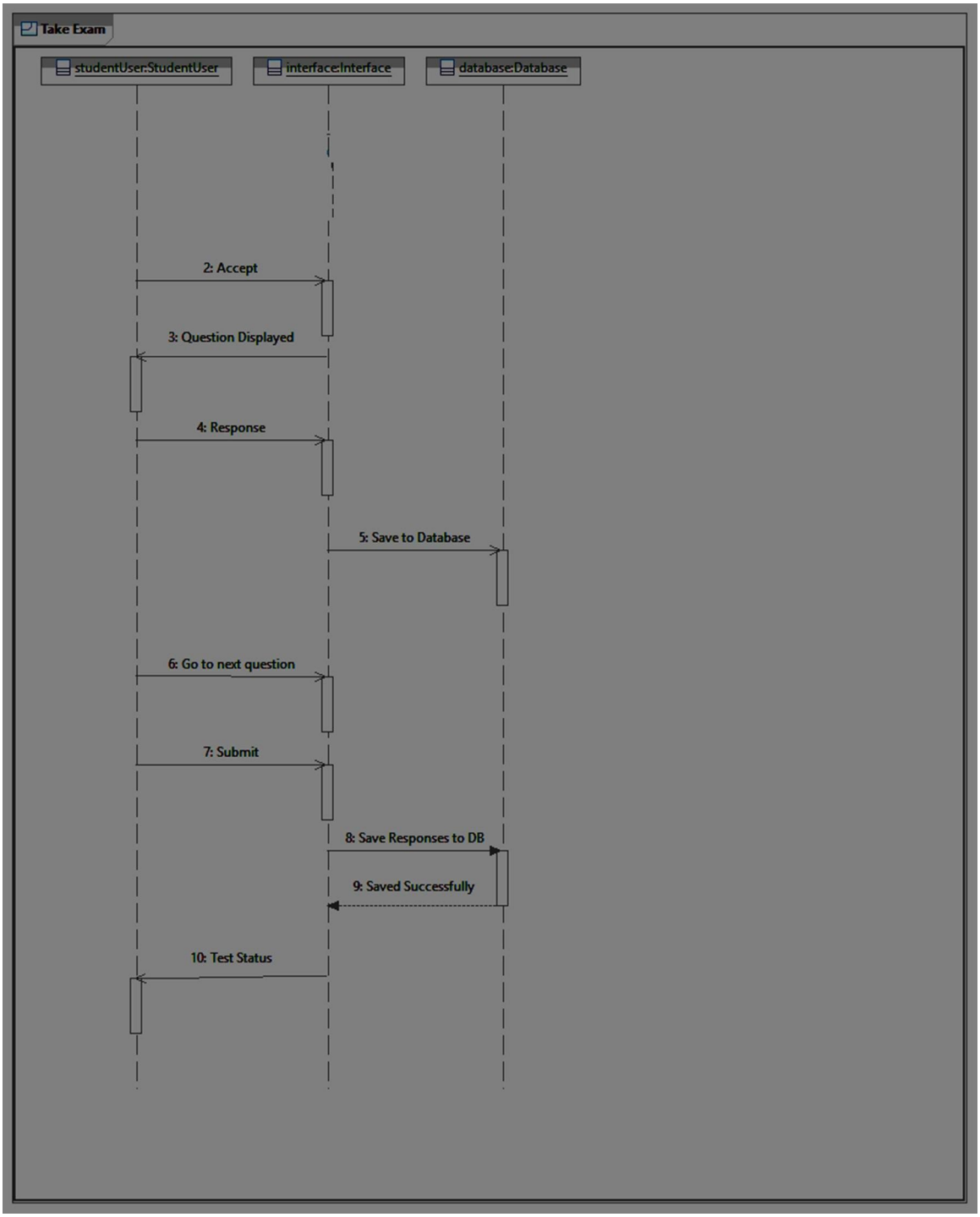
Login:



Add Student:

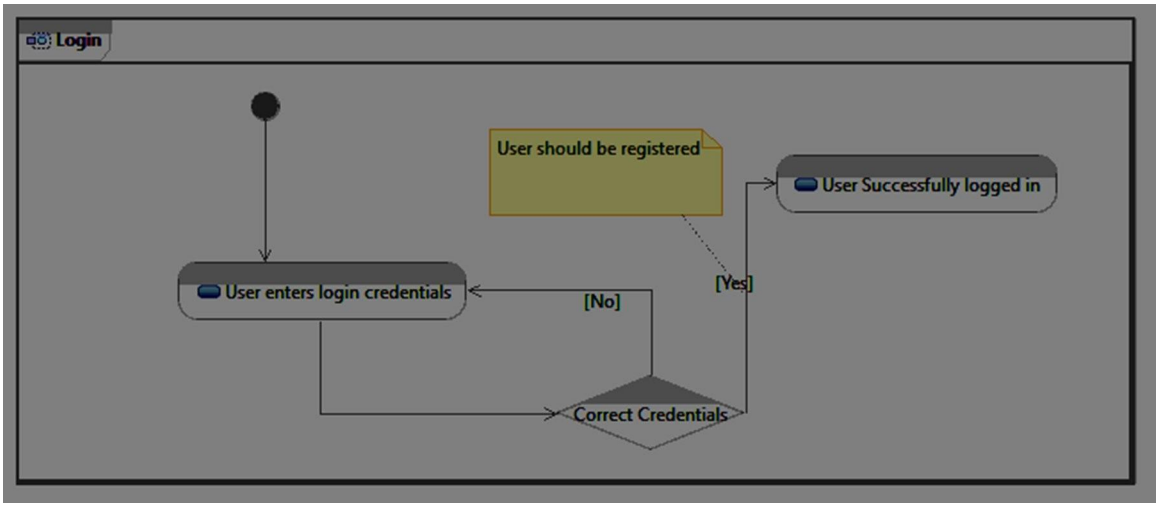

AddStudent
adminUser:AdminUser
interface:Interface
database:Database
1: Display form
2: Fill Form
3: Check if username exists in DB
4: Exist/Not Exist
5: if not exist, add to database
6: Status

Take Exam:

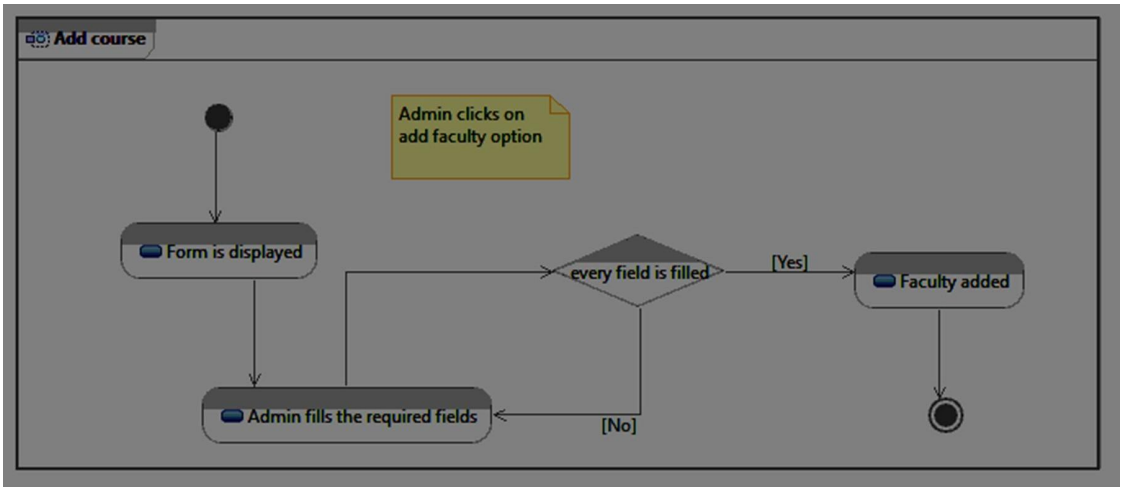

Take Exam
studentUser:StudentUser
interface:Interface
database:Database
2: Accept
3: Question Displayed
4: Response
5: Save to Database
6: Go to next question
7: Submit
8: Save Responses to DB
9: Saved Successfully
10: Test Status

Activity Diagrams:

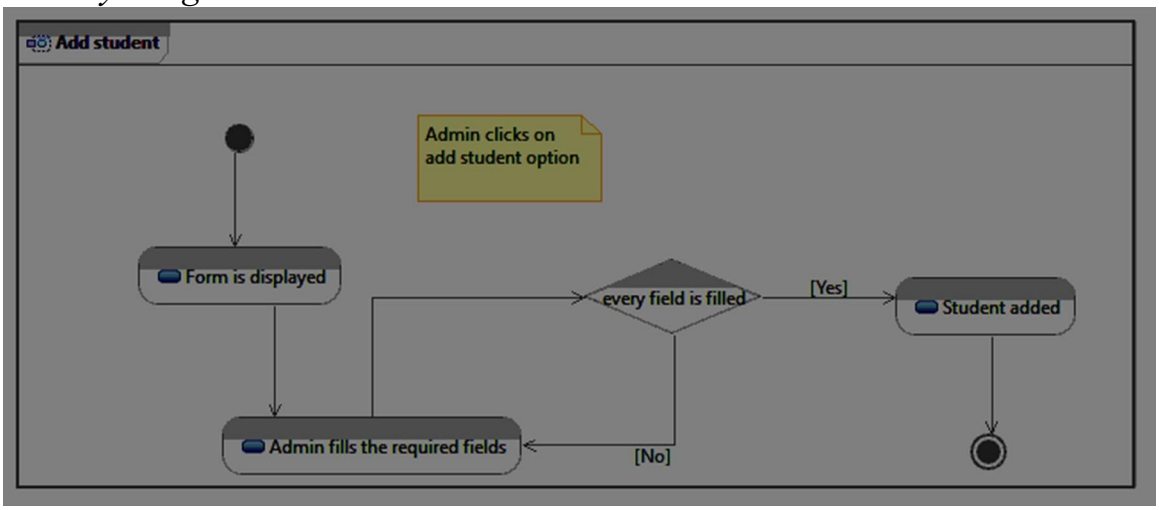
Activity Diagram: Login



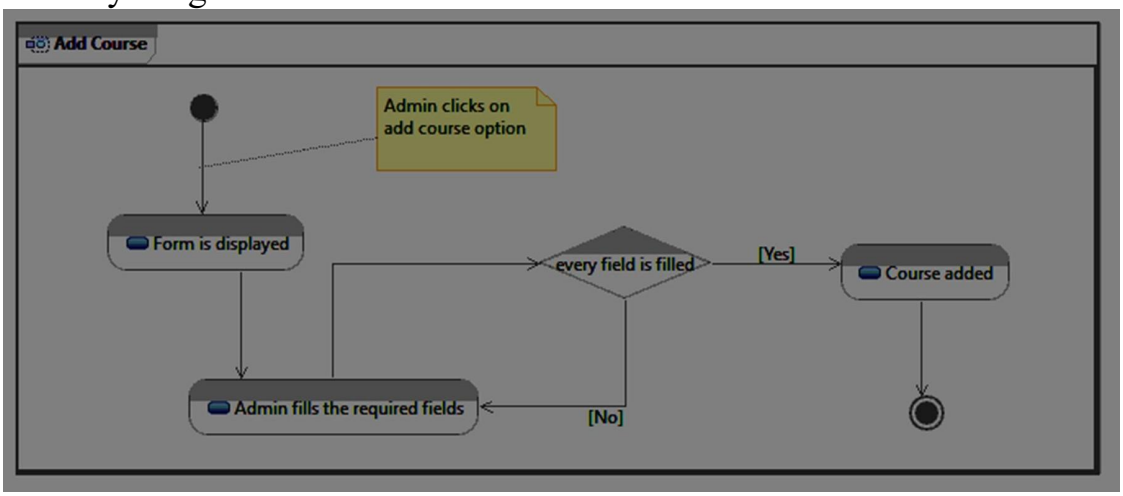
Activity Diagram: Add Faculty



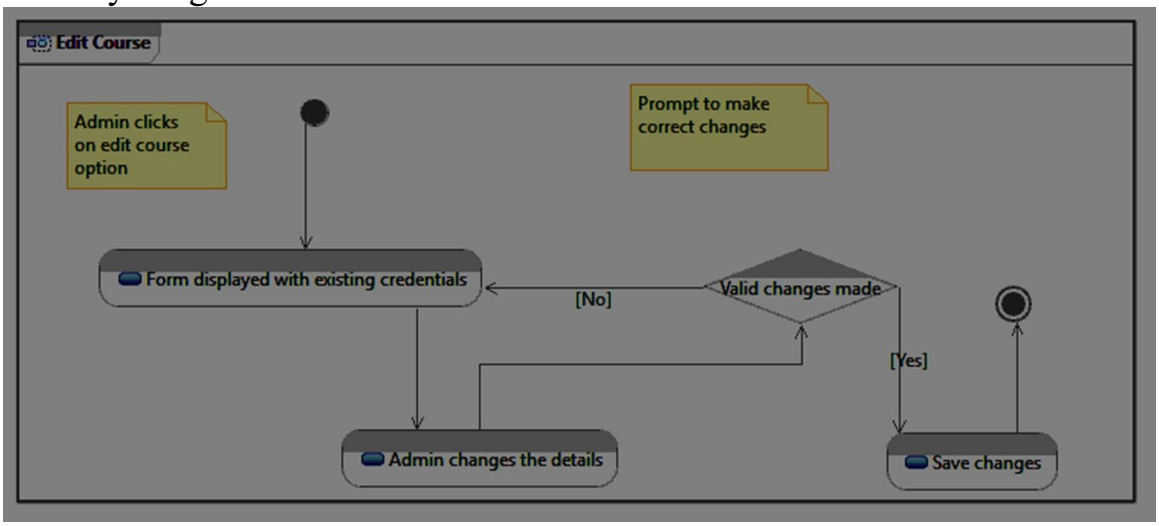
Activity Diagram: Add Student



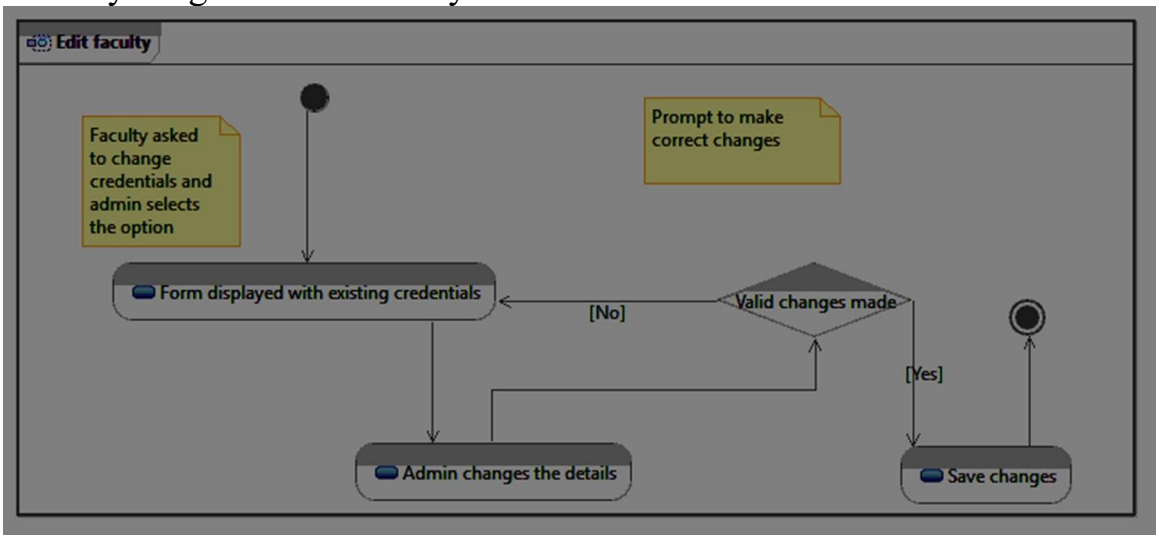
Activity Diagram: Add Course



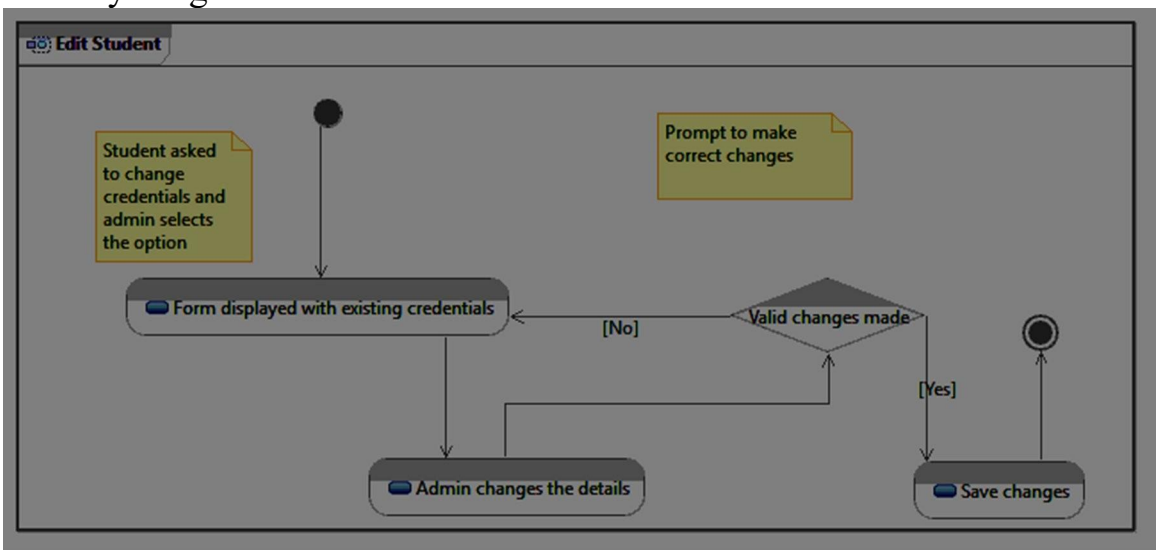
Activity Diagram: Edit Course



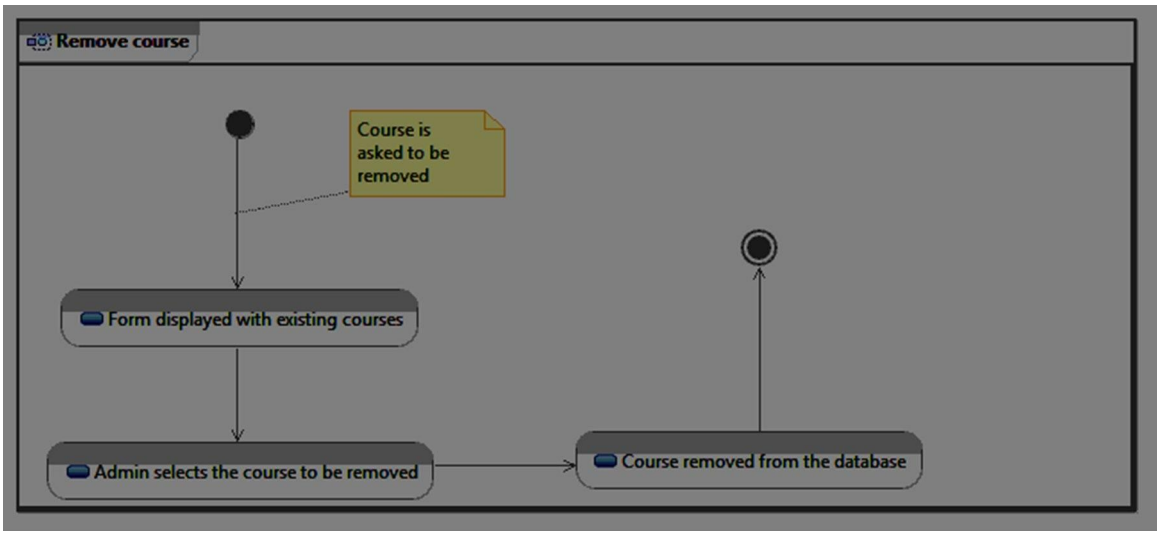
Activity Diagram: Edit Faculty



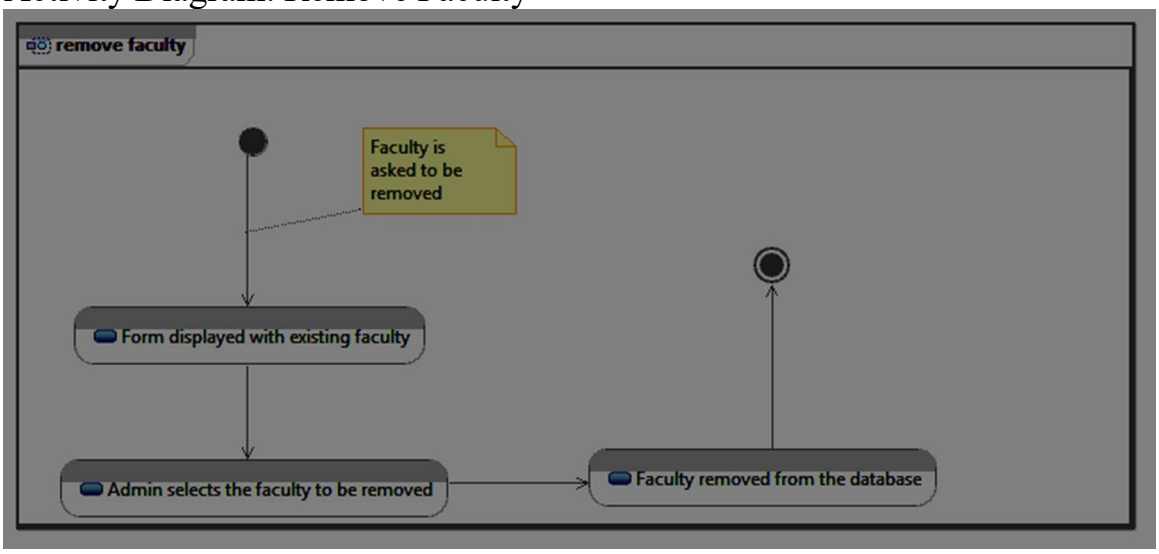
Activity Diagram: Edit Student



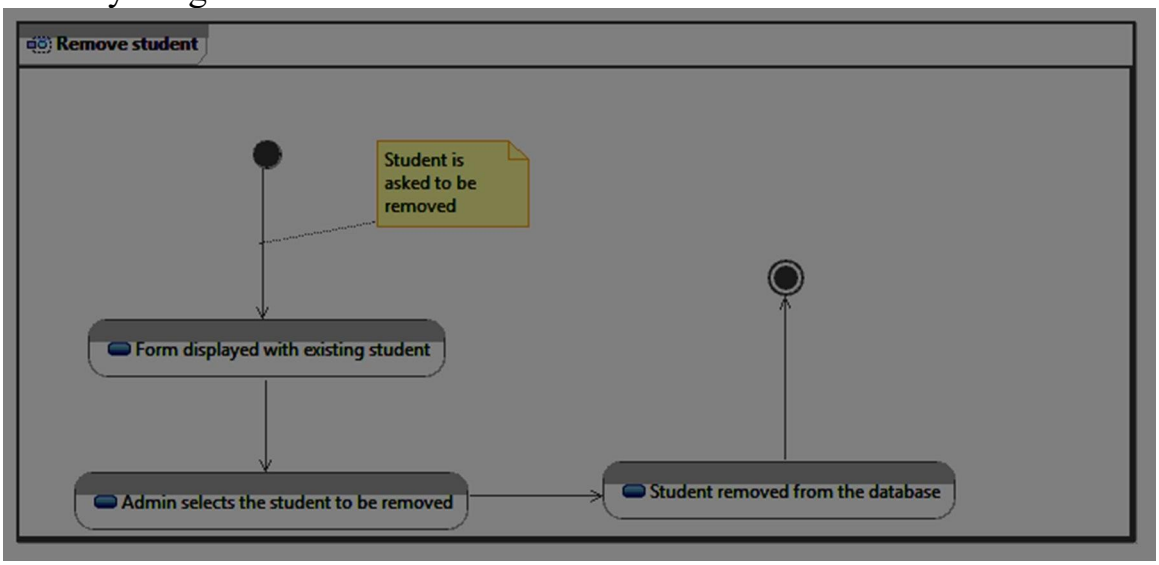
Activity Diagram: Remove Course



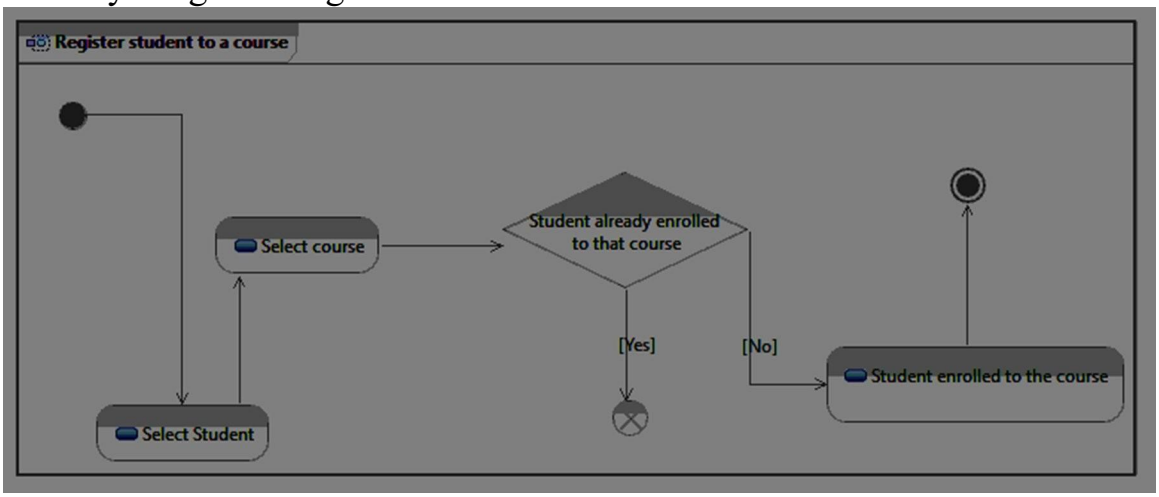
Activity Diagram: Remove Faculty



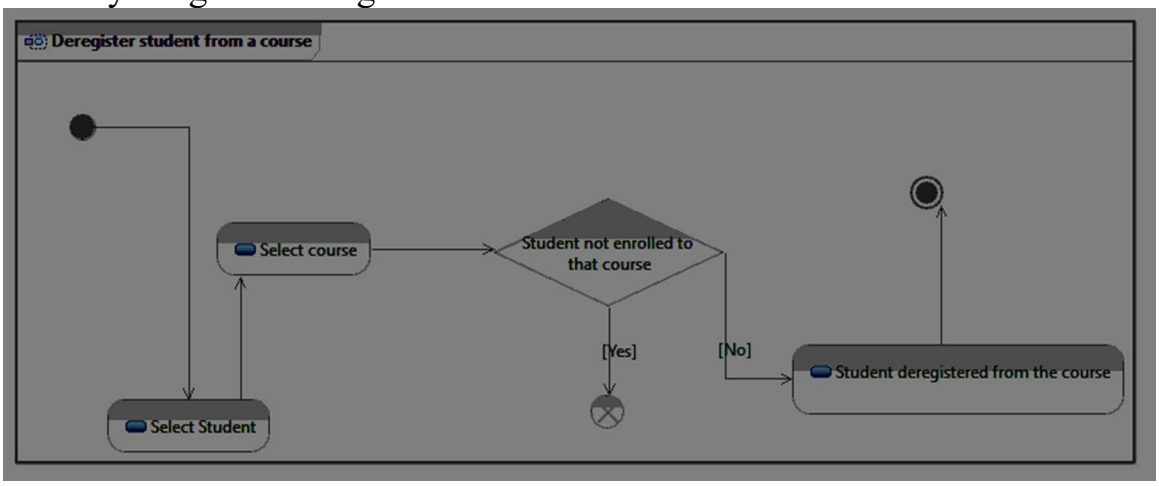
Activity Diagram: Remove Student



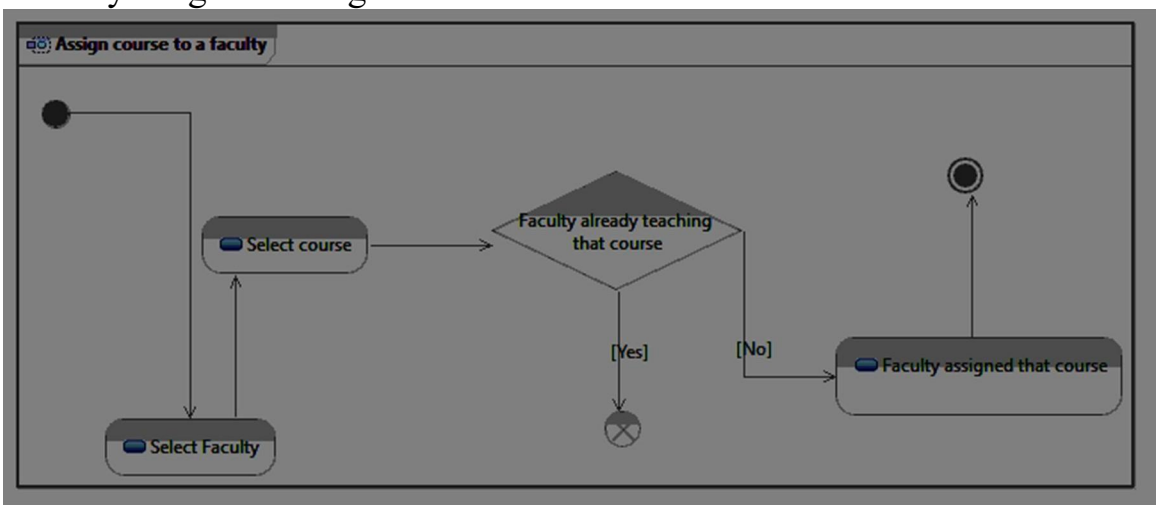
Activity Diagram: Register Student



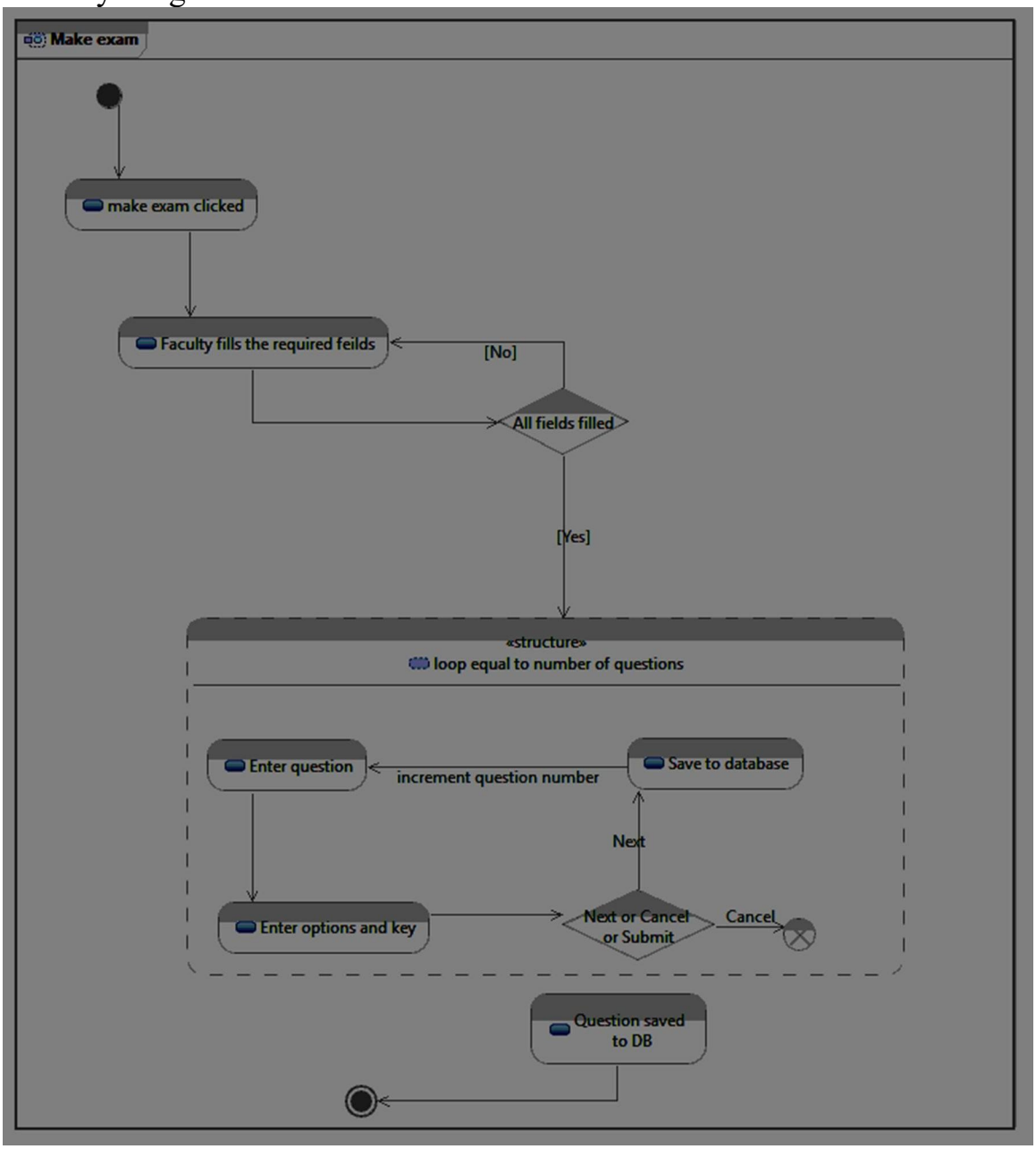
Activity Diagram: Deregister Student



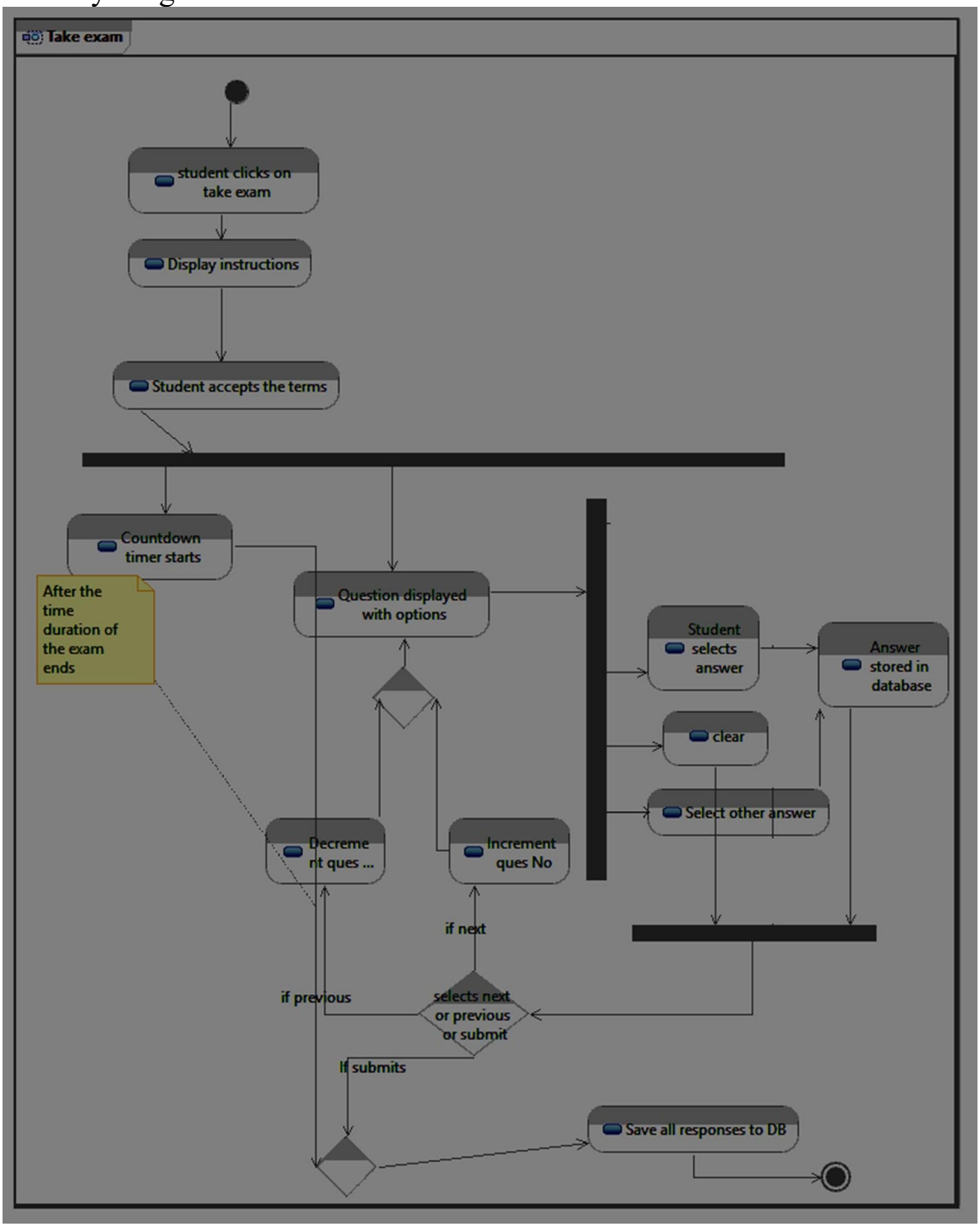
Activity Diagram: Assign Course



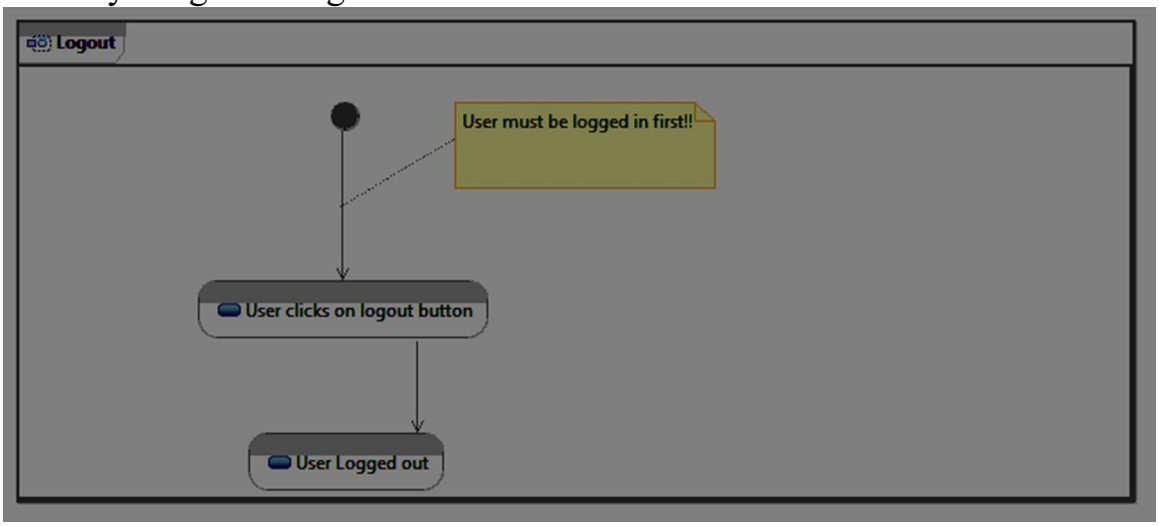
Activity Diagram: Make Exam



Activity Diagram: Take Exam



Activity Diagram: Logout



2.3 User Characteristics

The users are expected to be familiar with basic internet functionalities. Faculty and student users are expected to be able to use email. They should also be familiar with objective examination system.

All users must have valid LDAP credentials.

2.4 Non-Functional Requirements

A high speed internet connection is required to ensure smooth examination procedure. Latest version of a compatible browser is necessary. A server to host the web application and store data is required. It should be able to handle the number of users in the institute.

3.0. Requirements Specification

3.1 Functional Requirements

3.1.1 Add Student

Use Case Name	Add Student
Trigger	Admin user clicks on add Student option
Precondition	The Student is enrolled in the institute. The User is logged in as Admin
Basic Path	1. Admin enters the details of the Student into the system 2. Admin saves the details of the new user
Alternative Paths	
Postcondition	The new Student is added to the database
Exception Paths	The Admin may cancel the Student addition at any time
Other	None

3.1.2 Remove Student

Use Case Name	Remove Student
Trigger	Admin user clicks on remove Student option
Precondition	The Student has passed out of the institute or is not eligible for examinations due to other reasons
Basic Path	1. Admin enters the details of Student to be removed 2. Admin deletes the user
Alternative Paths	In step 2, if student does not already exist in database, then he is prompted to reenter the details of the Student
Postcondition	The Student is removed from the database
Exception Paths	The Admin may cancel the Student removal at any time
Other	None

2.1.3 Edit Student

Use Case Name	Edit Student
Trigger	Admin user clicks on edit student option
Precondition	The Student has requested for editing his profile details and the same is verified by competent authority.
Basic Path	1. Admin modifies the details of Student to be edited 2. Admin edits the user
Alternative Paths	In step 2, if entered student doesn't exist in database, then he is prompted to reenter the username of the student
Postcondition	The Student details are edited
Exception Paths	The Admin may cancel the Student edit at any time
Other	None

3.1.4 Add Faculty

Use Case Name	Add Faculty
Trigger	Admin user clicks on add Faculty option
Precondition	The Faculty is added to the institute. The User is logged in as Admin
Basic Path	1. Admin enters the details of the Faculty into the system 2. Admin saves the details of the new user
Alternative Paths	
Postcondition	The new Faculty is added to the database
Exception Paths	The Admin may cancel the Faculty addition at any time
Other	None

3.1.5 Remove Faculty

Use Case Name	Remove Faculty
Trigger	Admin user clicks on remove Faculty option
Precondition	The Faculty has left the institute
Basic Path	1. Admin enters the details of Faculty to be removed 2. Admin deletes the user
Alternative Paths	In step 2, if Faculty doesn't exist in database, then he is prompted to reenter the details of the Faculty
Postcondition	The Faculty is removed from the database
Exception Paths	The Admin may cancel the Faculty removal at any time
Other	None

2.1.6 Edit Faculty

Use Case Name	Edit Faculty
Trigger	Admin user clicks on edit Faculty option
Precondition	The Faculty has requested for editing his profile details and the same is verified by competent authority.
Basic Path	1. Admin enters the details of Faculty to be edited 2. Admin edits the user
Alternative Paths	In step 2, if entered Faculty doesn't already exist in database, then he is prompted to reenter the details of the Faculty
Postcondition	The Faculty details are edited
Exception Paths	The Admin may cancel the Faculty edit at any time
Other	None

3.1.7 Add Course

Use Case Name	Add Course
Trigger	Admin user clicks on add Course option
Precondition	A new Course is added to the institute
Basic Path	1. Admin enters the details of the Course into the system 2. Admin saves the details of the new Course
Alternative Paths	In step 2, if Admin does not verify the details, then he is prompted to reenter the details of the Course
Postcondition	The new Course is added to the database
Exception Paths	The Admin may cancel the Course addition at any time
Other	None

3.1.8 Remove Course

Use Case Name	Remove Course
Trigger	Admin user clicks on remove Course option
Precondition	The Course is removed from the institute
Basic Path	1. Admin enters the details of Course to be removed 2. Admin deletes the Course
Alternative Paths	In step 2, if Course doesn't already exist in database, then he is prompted to reenter the details of the Course
Postcondition	The Course is removed from the database
Exception Paths	The Admin may cancel the Course removal at any time
Other	None

3.1.9 Edit Course

Use Case Name	Edit Course
Trigger	Admin user clicks on edit Course option
Precondition	The course details are edited by the institute
Basic Path	1. Admin enters the details of Course to be edited 2. Admin edits the Course
Alternative Paths	In step 2, if entered course is not already present in database, then he is prompted to reenter the details of the student
Postcondition	The Course details are edited
Exception Paths	The Admin may cancel the Course edit at any time
Other	None

3.1.10 Enroll Student in Course

Use Case Name	Enroll Student in Course
Trigger	Admin user clicks on Enroll Student option
Precondition	Student enrolled in course as per academic details. Both Course and Student must be already added
Basic Path	1. Admin selects the student 2. Admin enters the course to which Student is to be added 3. Admin enrolls the student
Alternative Paths	In step 3, if entered Student or Course don't already exist in database, then he is prompted to reenter the correct details
Postcondition	The Student is enrolled in the course
Exception Paths	The Admin may cancel the enrollment process at any time
Other	None

3.1.11 Deregister Student from Course

Use Case Name	Deregister Student from Course
Trigger	Admin user clicks on Deregister Student from course option
Precondition	The Student is enrolled in the institute. The User is logged in as Admin
Basic Path	1. Admin selects the student. 2. Admin selects the course from the list of enrolled courses. 3. Admin deregisters the student from that course.
Alternative Paths	In step 3, if Student or Course don't exist in database, then he is prompted to reenter the details.
Postcondition	The new Student is deregistered from the particular course.
Exception Paths	The Admin may cancel the Student deregistration at any time
Other	None

3.1.12 Add faculty to course

Use Case Name	Add faculty to Course
Trigger	Admin user clicks on Add Faculty option
Precondition	Faculty has decided to take the course. Both Course and Faculty must be already added
Basic Path	1. Admin selects the Faculty 2. Admin enters the course to which Faculty is to be added 3. Admin adds the Faculty
Alternative Paths	In step 3, if Faculty or Course don't already exist in database, then he is prompted to reenter the correct details
Postcondition	The Faculty is added to the course
Exception Paths	The Admin may cancel the process at any time
Other	None

3.1.13 Create new exam

Use Case Name	Create new exam
Trigger	Faculty selects the create exam option
Precondition	Faculty teaches at least one course in that institute for which he/she can make the exam paper.
Basic Path	1. Faculty selects the option of creating a new exam. 2. Faculty enters the exam_code for which the exam is being made. 3. Faculty fills all the relevant details like exam name, time limit. 4. He/ She enters all the question along with four options and a key (correct answer). 5. Faculty verifies and finalizes the question paper.
Alternative Paths	If course doesn't exist in database already, he/ she is prompted to reenter.
Postcondition	A question paper is set for the course taught by the faculty.
Exception Paths	The faculty can cancel the course of action at any point of time.
Other	None

3.1.18 View Performance

Use Case Name	View Performance
Trigger	Student selects the option to display his/her performance in past exams.
Precondition	Student must be logged in and must be enrolled in at least one course.
Basic Path	1. Student clicks on View Performance option. 2. Student selects a course from a list of courses displayed.
Alternative Paths	If no exam has happened yet then the same will be showed.
Postcondition	Performance of the student in past exams is displayed on the screen
Exception Paths	The student can cancel the course of action at any point of time.
Other	None

3.1.19 Take New Exam

Use Case Name	Take New Exam
Trigger	Student selects the option to take an exam.
Precondition	An exam must have already been created by the faculty of the course the student has selected.
Basic Path	1. A student clicks on the Take Exam option. 2. If an exam is existing already, then student enters it's code. 3. The timer will start and the list of questions will appear sequentially and the student can select an answer, clear the response or move backward/ forward. 4. After finishing the exam, the student has to click the Submit button. 5. If the time finishes then Submit button is automatically triggered and the Take Exam option of that particular exam disappears from Student profile. 6. The responses of the student is saved in the system.
Alternative Paths	Step 2 – If any exam is not yet made, then the same will be shown in the profile. Step 6 - If Student decides to submit his response before time finishes, skip to Step 7.
Postcondition	The student is able to take the respective exam successfully and his performance will be stored.
Exception Paths	
Other	None

3.1.20 Login

Use Case Name	Login
Trigger	User fills his credentials in Login form
Precondition	None
Basic Path	1. User fills the information in Login form 2. Server side verification of filled details happen and upon successful signup, the user is directed to his/her profile
Alternative Paths	Step 2 – If invalid user details are submitted, the Login form is rendered again
Postcondition	The user is directed to his/her profile
Exception Paths	The user can cancel the Login by exiting the browser window
Other	None

3.1.21 Logout

Use Case Name	Logout
Trigger	User clicks logout button
Precondition	User must be logged in
Basic Path	1. User clicks logout button 2. User is logged out
Alternative Paths	None
Postcondition	None
Exception Paths	None
Other	None

3.2 Detailed Non-Functional Requirements

A high speed internet connection is required to ensure smooth examination procedure. Latest version of a compatible browser is necessary. A server to host the web application and store data is required. It should be able to handle the number of users in the institute.

3.3 Logical Structure of the Data

MongoDB, a NoSQL database is used for storing all application data.

The collections in the database are:

Admin
Attributes: username, password

Students
Attributes: username, password, name, rollno, course_list

Faculties
Attributes: username, password, name, course_list

Courses
Attributes: courseid, coursename

Exams
Attributes: exam_name, exam_code, duration_hours, duration_minutes, course_code, faculty_username

Responses
Attributes: username, exam_code, response