

STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

18CSC202J/ 18AIC203J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY

(2018 Regulation)

II Year/ III Semester

Academic Year: 2022 -2023

By

ANIKET RAJ SINGH (RA2111003011474)

Under the guidance of

Dr. A. DEVIPRIYA

Assistant Professor

Department of Computing Technologies



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

NOVEMBER 2022

BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled “**STUDENT MANAGEMENT SYSTEM**” is the bonafide work of **ANIKET RAJ SINGH(RA2111003011474)**

who undertook the task of completing the project within the allotted time.

Signature of the Guide

Dr. A. Devipriya

Assistant Professor

Department: Computing Technologies

SRM Institute of Science and Technology

Signature of the II Year Academic Advisor

Professor and Head

Department: Computing Technologies

SRM Institute of Science and Technology

About the course:-

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

Objectives:

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

Course Learning Rationale (CLR): The purpose of learning this course is to:

- 1.Utilize class and build domain model for real-time programs
- 2.Utilize method overloading and operator overloading for real-time application development programs
- 3.Utilize inline, friend and virtual functions and create application development programs
- 4.Utilize exceptional handling and collections for real-time object-oriented programming applications
- 5.Construct UML component diagram and deployment diagram for design of applications
- 6.Create programs using object-oriented approach and design methodologies for real-time application development

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

- 1.Identify the class and build domain model
- 2.Construct programs using method overloading and operator overloading
- 3.Create programs using inline, friend and virtual functions, construct programs using standard templates
- 4.Construct programs using exceptional handling and collections
- 5.Create UML component diagram and deployment diagram
- 6.Create programs using object oriented approach and design methodologies

Table 1: Rubrics for Laboratory Exercises

(Internal Mark Splitup:- As per Curriculum)

CLAP-1	5=(2(E-lab Completion) + 2(Simple Exercises) (from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-2	7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-3	7.5=(2.0(E-lab Completion(80 Pgms)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	2 Mark - E-lab Completion 80 Program Completion from 10 Session (Each session min 8 program) 2 Mark - Code to UML conversion GCR Exercises 3.5 Mark - Hacker Rank Coding challenge completion
CLAP-4	5= 3 (Model Practical) + 2(Oral Viva)	<ul style="list-style-type: none"> • 3 Mark – Model Test • 2 Mark – Oral Viva
Total	25	

COURSE ASSESSMENT PLAN FOR OODP LAB

S.No	List of Experiments	Course Learning Outcomes (CLO)	Blooms Level	PI	No of Programs in each session
1.	Implementation of I/O Operations in C++	CLO-1	Understand	2.8.1	10
2.	Implementation of Classes and Objects in C++	CLO-1	Apply	2.6.1	10
3.	To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram.	CLO-1	Analysis	4.6.1	Mini Project Given
4.	Implementation of Constructor Overloading and Method Overloading in C++	CLO-2	Apply	2.6.1	10
5.	Implementation of Operator Overloading in C++	CLO-2	Apply	2.6.1	10
6.	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and	CLO-2	Analysis	4.6.1	Mini Project Given
7.	Implementation of Inheritance concepts in C++	CLO-3	Apply	2.6.1	10
8.	Implementation of Virtual function & interface concepts in C++	CLO-3	Apply	2.6.1	10
9.	Using the identified scenarios in your project, draw relevant state charts and activity diagrams.	CLO-3	Analysis	4.6.1	Mini Project Given
10.	Implementation of Templates in	CLO-3	Apply	2.6.1	10
11.	Implementation of Exception of Handling in C++	CLO-4	Apply	2.6.1	10

12.	Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component	CLO-5	Analysis	4.6.1	Mini Project Given
13.	Implementation of STL Containers in C++	CLO-6	Apply	2.6.1	10
14.	Implementation of STL associate containers and algorithms in C++	CLO-6	Apply	2.6.1	10
15.	Implementation of Streams and File Handling in C++	CLO-6	Apply	2.6.1	10

LIST OF EXPERIMENTS FOR UML DESIGN AND MODELLING:

To develop a mini-project by following the exercises listed below.

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

Suggested Software Tools for UML:

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

ABSTRACT

Students form a main part of any institution that concerns with. But the institutions find it difficult to keep details of so many students of the organisation just in one stretch. It will involve a lot of pen paper work. Sometimes there will be some huge heap of files bundled up and kept together in some corner of the office. If you want any information regarding the particular student then it can be obtained by just entering the roll number or the name of the student to be searched. This student management system will make the work of storing the data in an organised way.

The student management system application will help in managing the student's reports, results and exams will become easier with one such system. It will also help in saving time and effort. The user interface must be user friendly and easy to understand. The information of the particular student will be obtained in just one mouse click. Some of the features that it can include are as follows:

Student database management: The details of the students of the organisations can be stored in the database with the use this application.

Results: The results of the students can also be accessed and stored through this application. Security: The data that will be disclosed will be more secure since there will be no access to the unknown users.

Performance: The performance of the students might be in curriculum as well as co-curriculum can also be stored through the use of this application.

One-click access: You will obtain the details of the students by entering his/her name or the roll number just in one click.

User interface: The user interface must be simple and easy to understand.

Personal details: All the personal details of the students can be obtained in just one mouse click.

Student Management System is software which is helpful for students as well as the school authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Student Management System deals with the various activities related to the students. In the Software we can register as a user and user has of two types, student and administrator. Administrator has the power to add new user and can edit and delete a user. A student can register as user and can add edit and delete his profile.

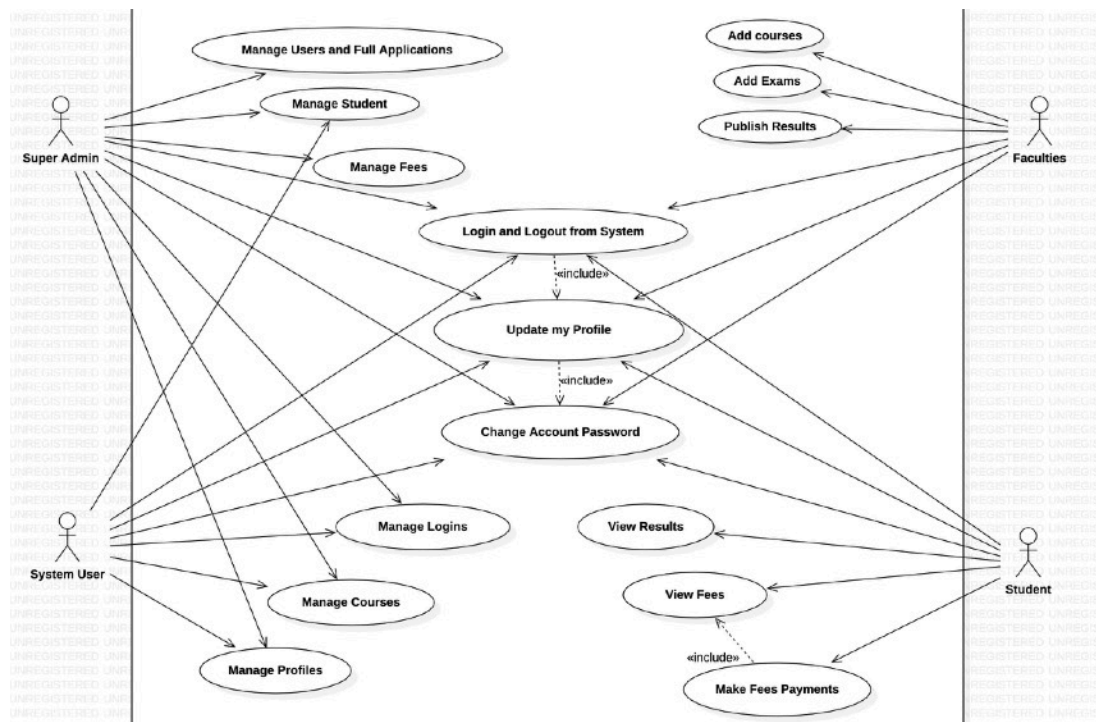
MODULE DESCRIPTION

Schools and Universities are the foundation of knowledge and an educational body on which students rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

Most schools and Universities count on an advanced software tool known as 'Student management System (SMS)' to keep all their student records and administrative operations including, examinations, attendance, and other activities.

Over the recent years, the performance and efficiency of the education industry have been enhanced by using the Student Management System. This tool has productively taken over the workload of the admin department with its well-organised, easy, and reliable online school management software.

Use case diagram with explanation

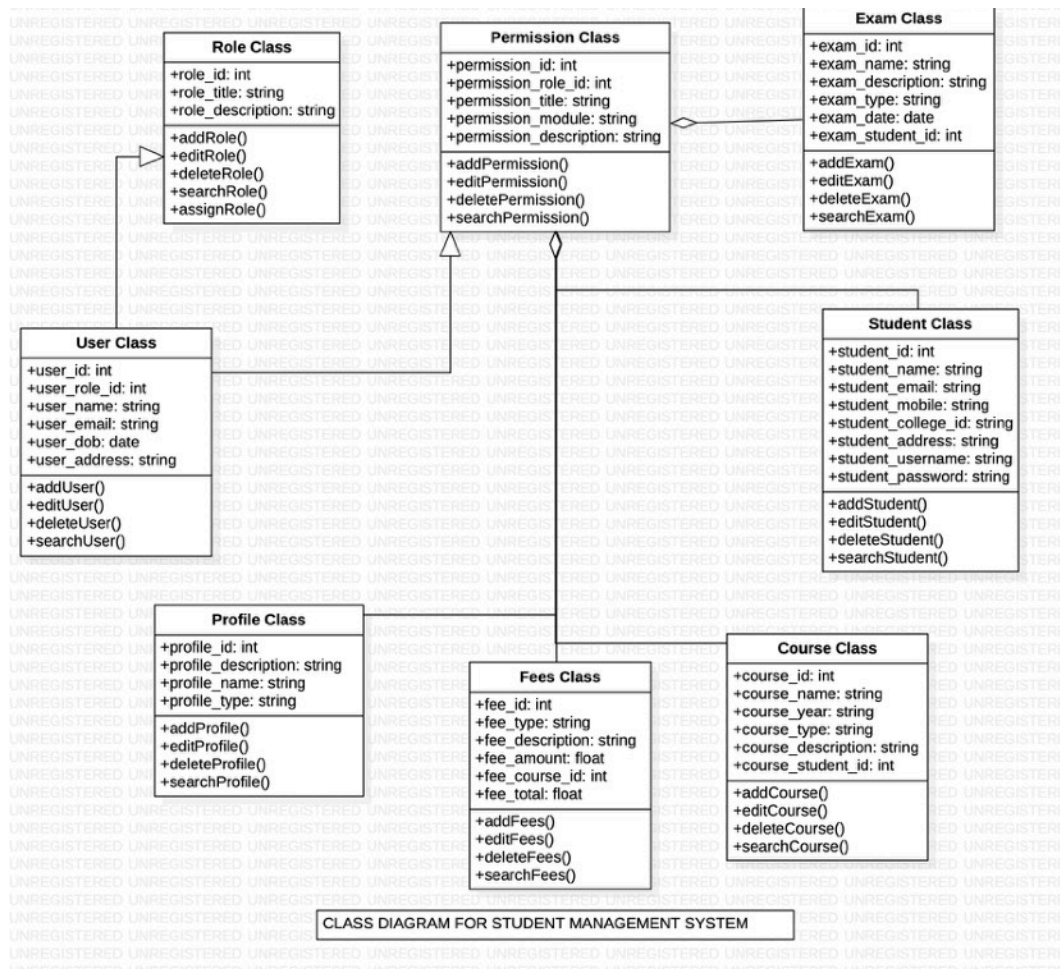


This Use Case Diagram is a graphic depiction of the interactions among the elements of Student Management System. It represents the methodology used in system analysis to identify, clarify, and organise system requirements of Student Management System. The main actors of Student Management System in this Use Case Diagram are: Super Admin, System User, Faculties, Student, who perform the different type of use cases such as Mange Student, Manage Fees, Manage Logins, Manage Profiles, Manage Courses, Manage Exams, Manage Users and Full Student Management System Operations. Major elements of the UML use case diagram of Student Management System are shown on the picture below.

The relationships between and among the actors and the use cases of Student Management System: Super Admin Entity : Use cases of Super Admin are Mange Student, Manage Fees, Manage Logins, Manage Profiles, Manage Courses, Manage Exams, Manage Users and Full Student Management System Operations

10 System User Entity : Use cases of System User are Mange Student, Manage Fees, Manage Logins, Manage Profiles, Manage Courses, Manage Exams
 Faculties Entity : Use cases of Faculties are Add Courses, Add Exams, Publish Results, View Students Entity : Use cases of Student are View Results, Views Fees, Make Fees Payments.

Class diagram with explanation



Student Management System Class Diagram describes the structure of a Student Management System classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the Student Management System are Student, Fees, Logins, Profiles, Courses, Exams. Classes of Student Management System Class Diagram:

Student Class : Manage all the operations of Student
 Fees Class : Manage all the operations of Fees
 Logins Class : Manage all the operations of Logins
 Profiles Class : Manage all the operations of Profiles
 Courses Class : Manage all the operations of Courses
 Exams Class : Manage all the operations of Exams

12Classes and their attributes of Student Management System Class Diagram:
 Student Attributes : student_id, student_college_id, student_name, student_mobile, student_email, student_username, student_password, student_address

Fees Attributes : fee_id, fee_course_id, fee_amount, fee_total, fee_payment, fee_type, fee_description

Logins Attributes : login_id, login_user_id, login_role_id, login_username, login_password, login_lastlogin

Profiles Attributes : profile_id, profile_name, profile_type, profile_description

Courses Attributes : course_id, course_student_id, course_registration, course_name, course_type, course_year, course_description

Exams Attributes : exam_id, exam_student_id, exam_roll_number, exam_date, exam_name, exam_type, exam_description

Classes and their methods of Student Management System Class Diagram:

Fees Methods : addFees(), editFees(), deleteFees(), updateFees(), saveFees(), searchFees()

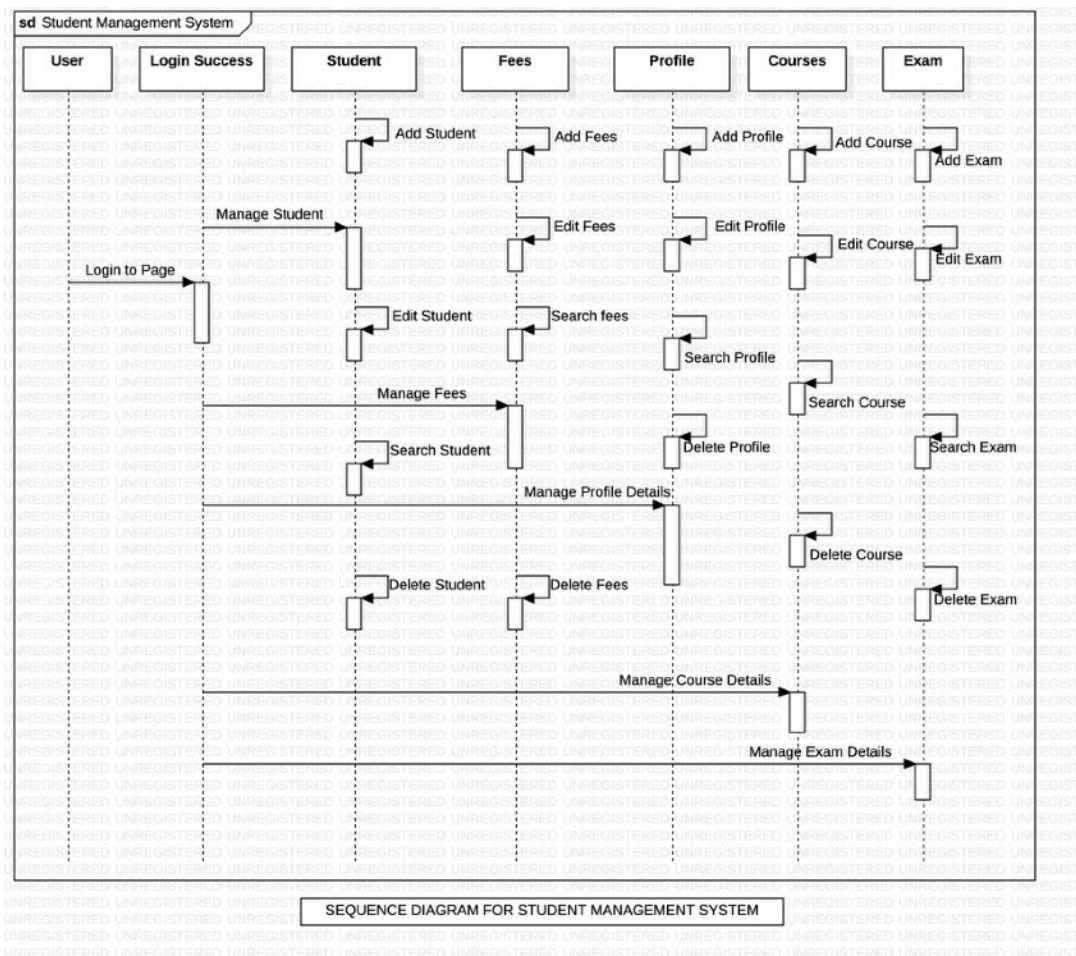
Logins Methods : addLogins(), editLogins(), deleteLogins(), updateLogins(), saveLogins(), searchLogins()

Profiles Methods : addProfiles(), editProfiles(), deleteProfiles(), updateProfiles(), saveProfiles(), searchProfiles()

Courses Methods : addCourses(), editCourses(), deleteCourses(), updateCourses(), saveCourses(), searchCourses()

Exams Methods : addExams(), editExams(), deleteExams(), updateExams(), saveExams(), searchExams().

Sequence diagram with explanation



This is the UML sequence diagram of Student Management System which shows the interaction between the objects of Profiles, Exams, Fees, Student, Courses. The instance of class objects involved in this UML Sequence Diagram of Student Management System are as follows:

Profiles Object Exams Object Fees Object Student Object Courses Object

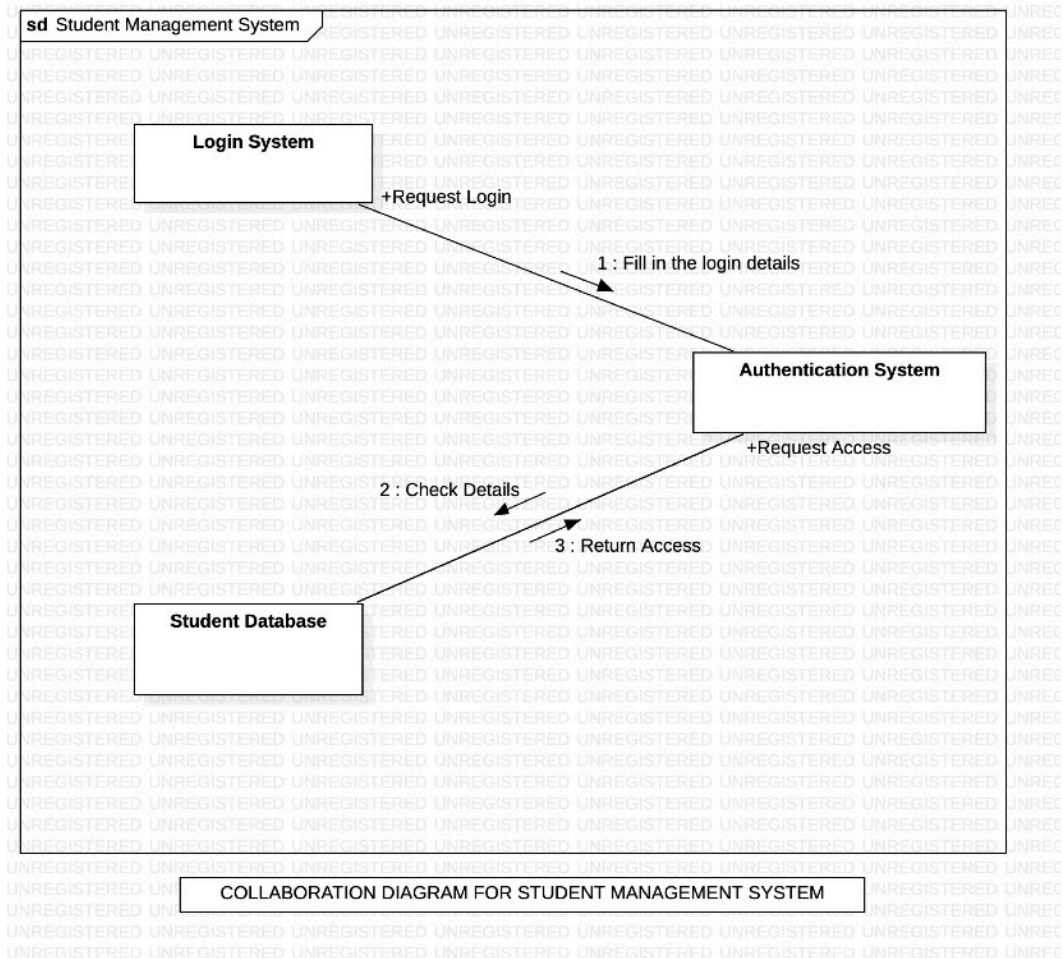
14Login Sequence Diagram Of Student Management System:

This is the Login Sequence Diagram of Student Management System, where admin will be able to login in their account using their credentials. After login user can manage all the operations on Fees, Profiles, Exams, Courses, Student. All the pages such as Exams, Courses, Student are secure and user can access these page after login. The diagram below helps demonstrate how the login page works in a Student Management System. The various objects in the Courses, Fees, Profiles, Exams, and Student page—interact over the course of the sequence, and user will not be able to access this page without verifying their identity.

This is the UML sequence diagram of Student Management System which shows the interaction between the objects of Profiles, Exams, Fees, Student, Courses. The instance of class objects involved in this UML Sequence Diagram of Student Management System are as follows:

Profiles Object Exams Object Fees Object Student Object Courses Object.

Communication diagram with explanation



Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction.

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. The four major components of a collaboration diagram are:

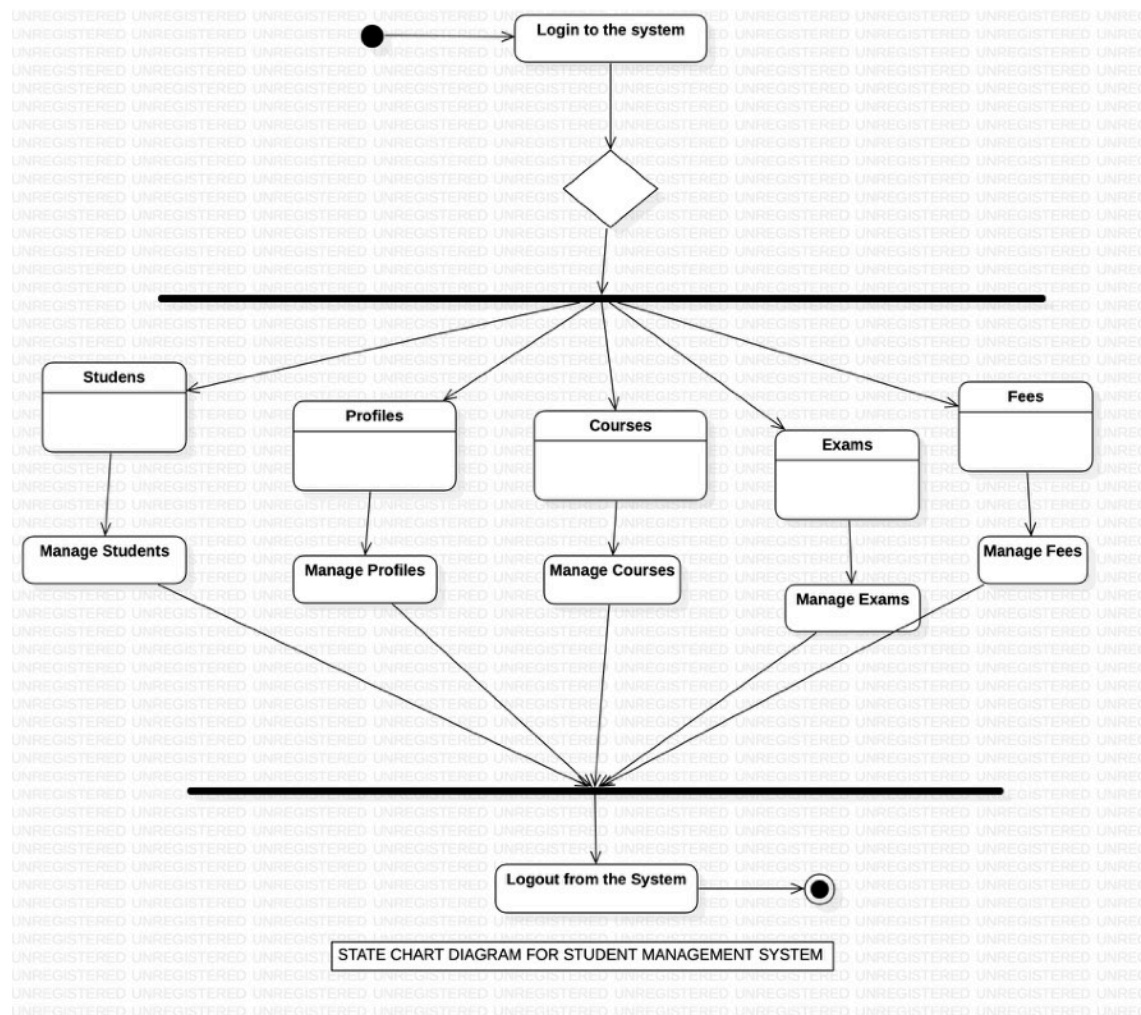
Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.

Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.

Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

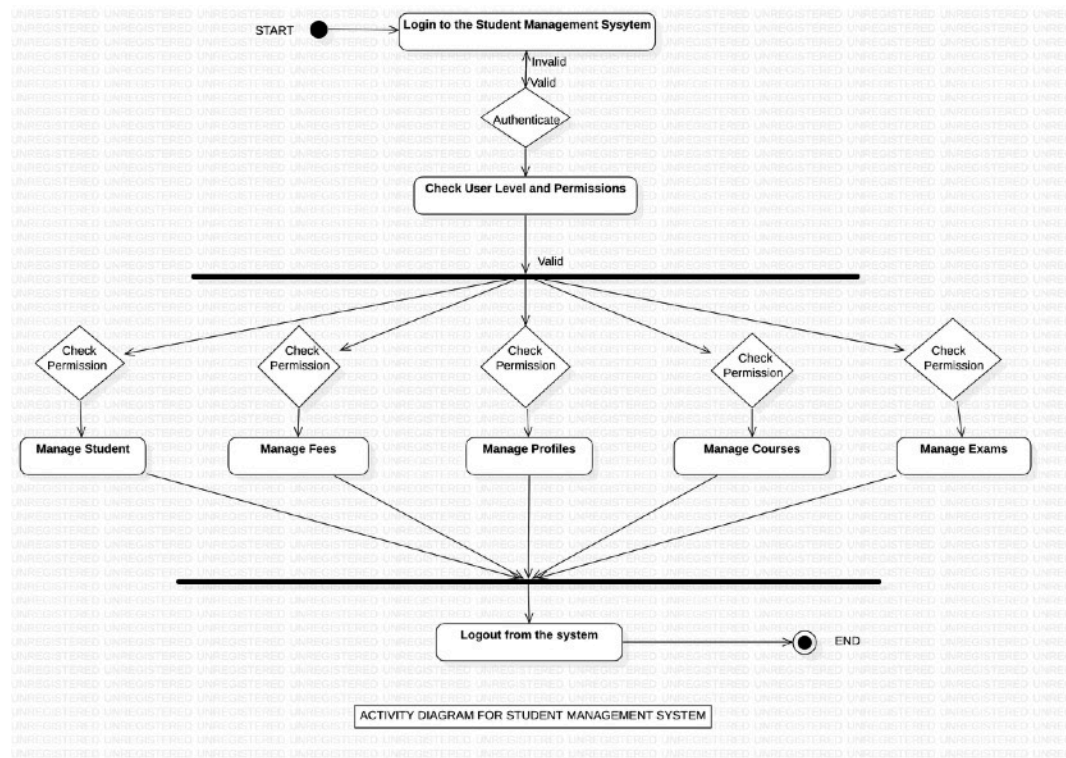
State chart diagram with explanation



State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

Here, Students, Profiles, Courses, Exams and Fees are in Composite states and Login, manage and Logout are in transition states.

Activity diagram with explanation



This is the Activity UML diagram of Student Management System which shows the flows between the activity of Fees, Profiles, Student, Courses, Exams. The main activity involved in this UML Activity Diagram of Student Management System are as follows:

Fees Activity

Profiles Activity

Student Activity

Courses Activity

Exams Activity

Features

Of The Activity UML Diagram Of Student Management System.

Admin and User can search Fees, view description of a selected Fees, add Fees, update Fees and delete Fees.

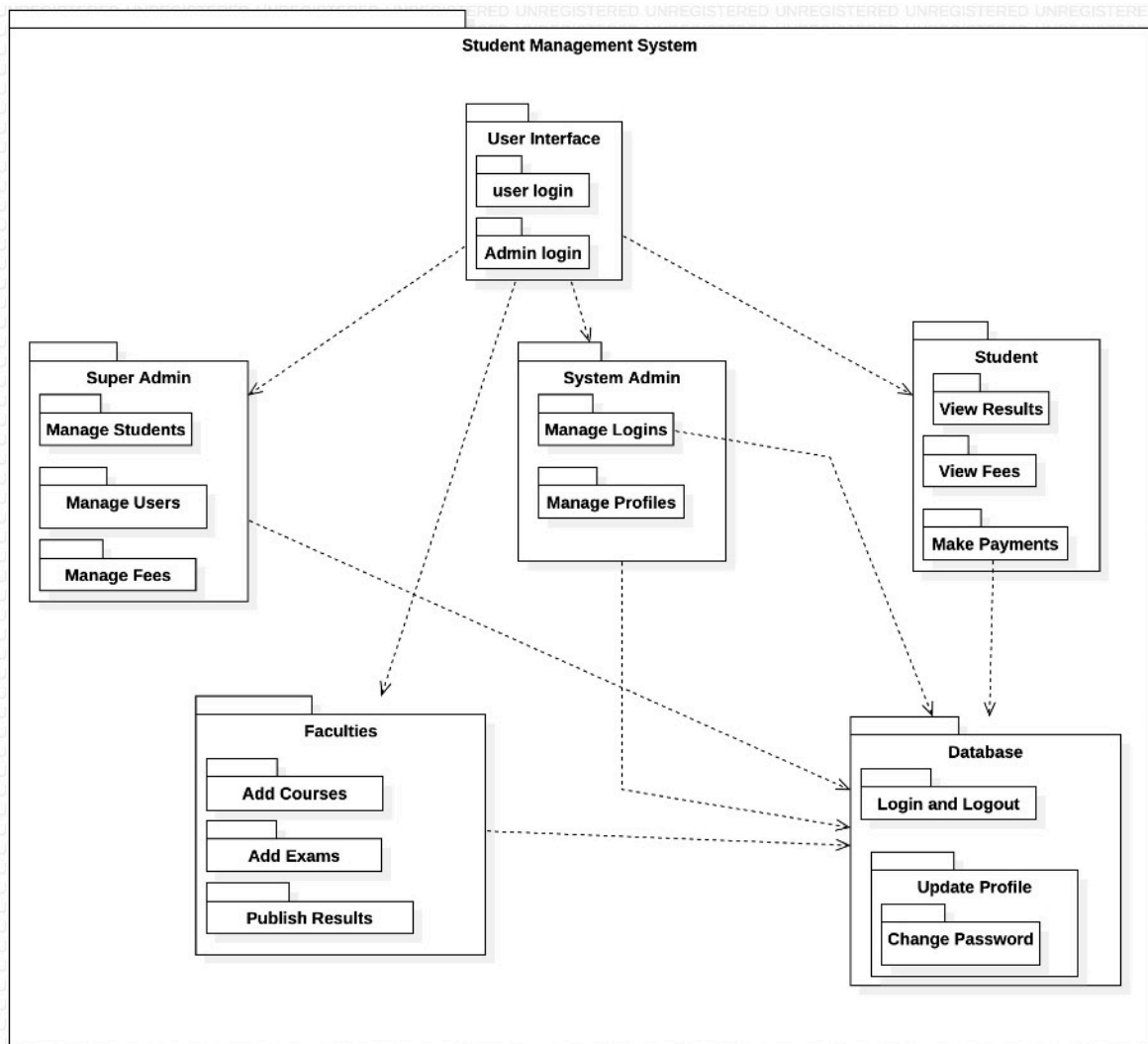
Its shows the activity flow of editing, adding and updating of Profiles

User will be able to search and generate report of Student, Courses, Exams

All objects such as (Fees, Profiles, Exams) are interlinked

Its shows the full description and flow of Fees, Courses, Exams, Student, Profiles.

Package diagram with explanation



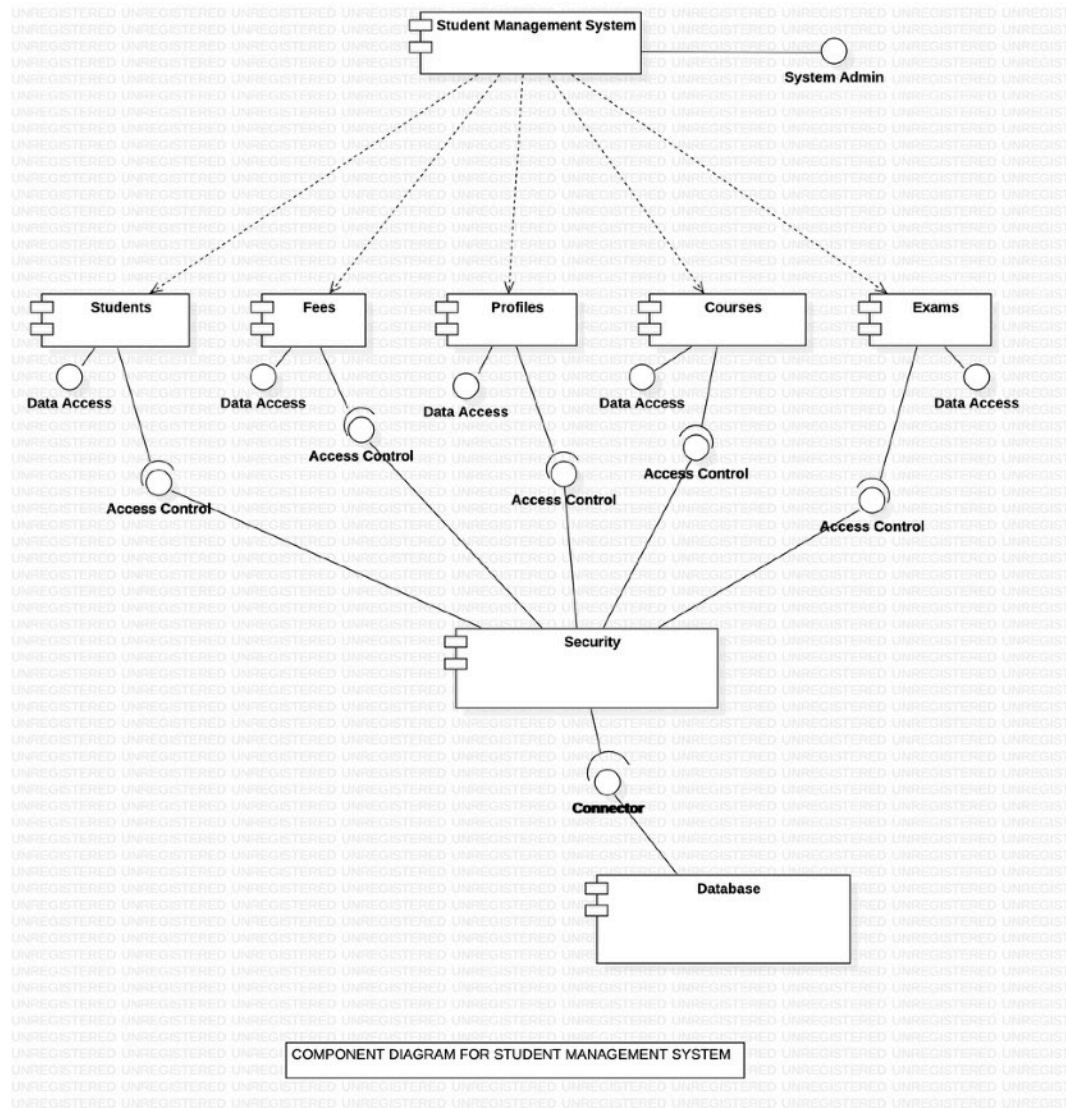
PACKAGE DIAGRAM FOR STUDENT MANAGEMENT SYSTEM

Package diagram of student management system are structural diagram that is usually accustomed alter advanced category diagrams and organise categories into packages. A package may be a assortment of connected UML components as well as diagrams, documents, classes, and events. apart from that, the package diagram offers valuable high- level visibility for giant comes and systems.

Here, Student Management System is the largest package which contains small packages - User Interface, Super Admin, System Admin, Student, Faculties, Database which contains further small packages as their functionalities.

All Packages inside the system have a dependent relationship with Database Package.

Component diagram with explanation



This is a Component diagram of Student Management System which shows components, provided and required interfaces, ports, and relationships between the Logins, Courses, Fees, Exams and Student.

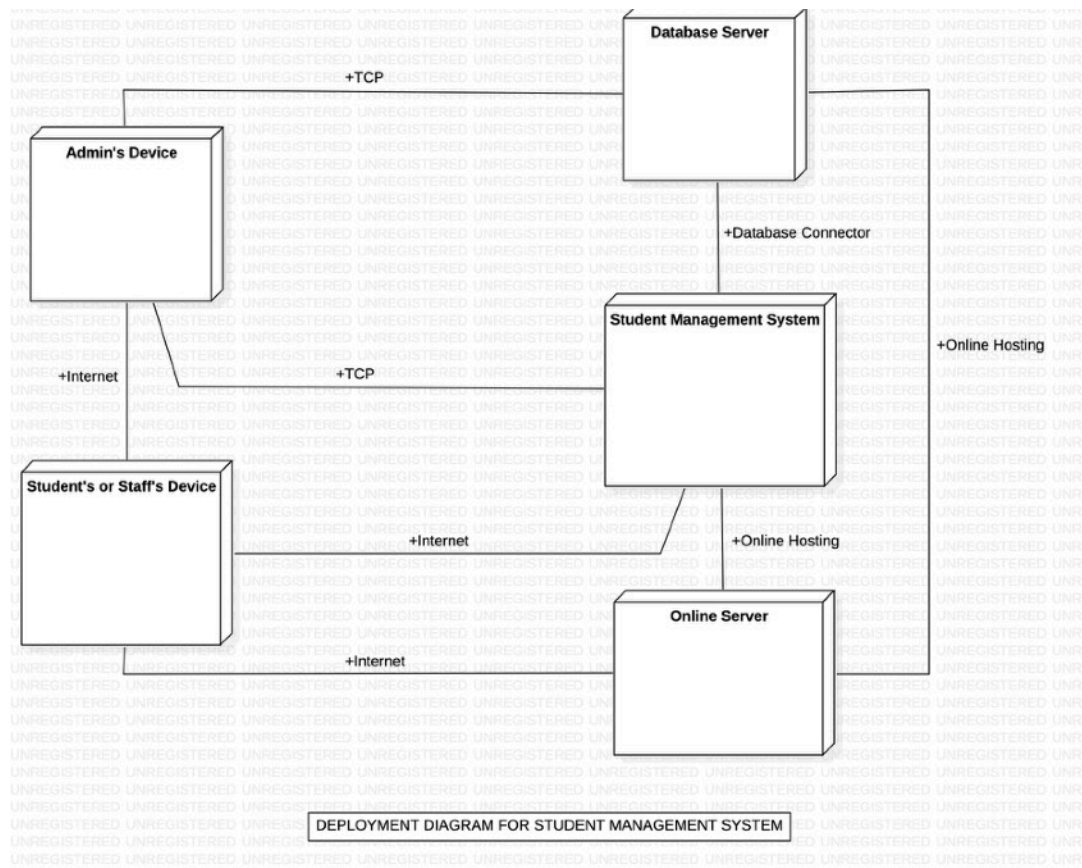
Student Management System UML component diagram, describes the organisation and wiring of the physical components in a system.

Components of UML Component Diagram of Student Management System: Logins Component, Courses Component, Fees Component, Exams Component and Student Component

Features of Student Management System Component Diagram:

You can show the models the components of Student Management System. Model the database schema of Student Management System and executables of an application of Student Management System Model the system's source code of Student Management System.

Deployment diagram with explanation



The Student Management System UML deployment diagram explains the sketch of the relationship between software and hardware. These hardware and software are labeled to clarify their part in the system's operation. They were represented by nodes and the connections were represented by labeled arrows. The deployment diagram shows the scenario when the system is deployed. It has 5 nodes represented with boxes and relationship connections. The nodes are the student management system, the browser, the employer's device, the employee's device, and the database (system server). The system server node contains a developed database that will hold the details of the system online.

For the connection, the system is connected to the server database using a private network which enables it to pass a connection to the devices and enable users to access the system and database. The employer and the employees then can communicate through the system.

Conclusion

All in all, the UML Diagrams work along to attain the foremost desired functions of a web looking System. All of those were designed to guide likewise as facilitate programmers and beginners of what ought to be the behaviour and structure of the web looking System.

By finishing all the given Diagrams, the web looking System development would terribly be abundant easier and a lot of possible. thus those UML diagrams got to show you and guide you thru your project development journey. you'll use all the given UML diagrams as your reference or have them for your project development. The ideas bestowed in UML Diagrams were all supported on-line looking System needs.

References

1. itsourcecode.com
2. [GeeksforGeeks](https://www.geeksforgeeks.org)
3. [Creatly.com](https://www.creatly.com)