DASS ASSIGNMENT-3

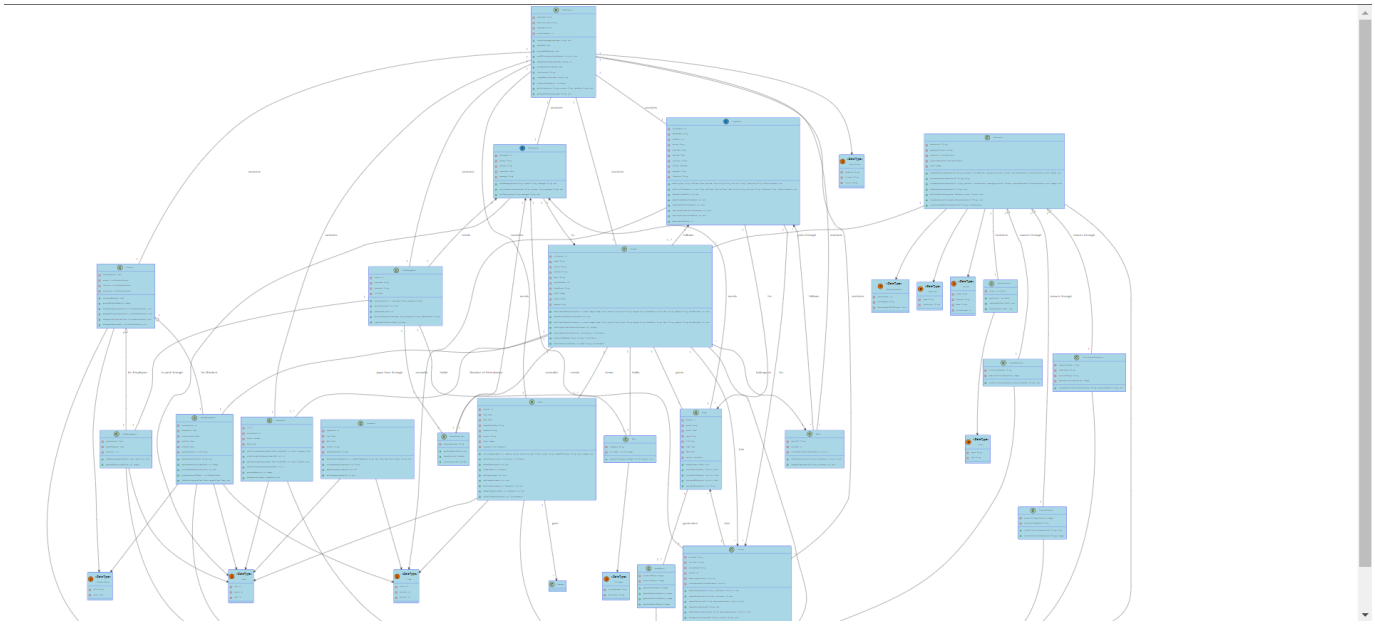
Team Members-

➢Mayank Mittal(2022101094)

➢Aditya Garg(2022101083)

UML Modelling Tool- PlantUML

Link for the class diagram:- Class Diagram.txt



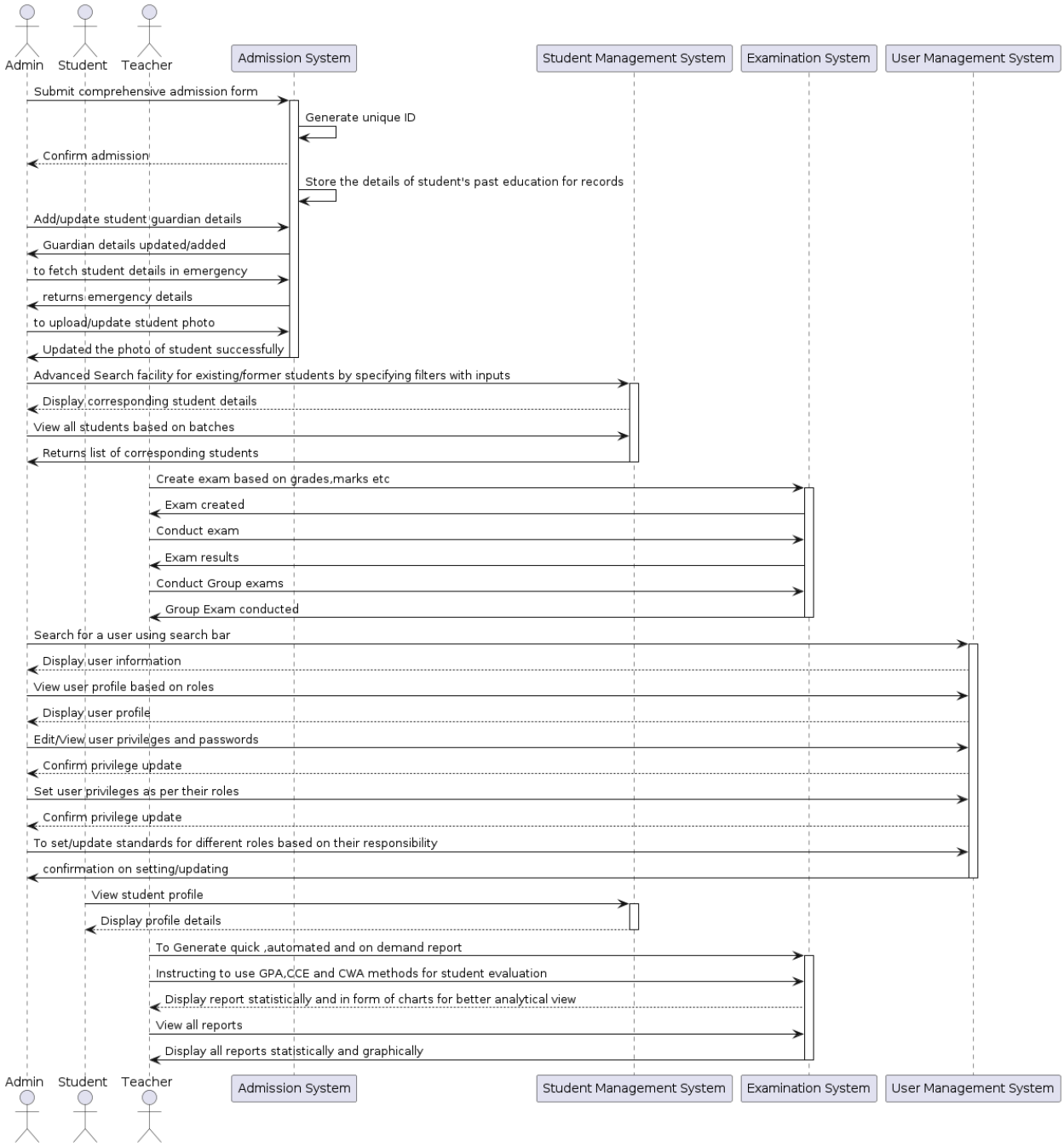
All the images are included in the doc as well as in the zip in png format.

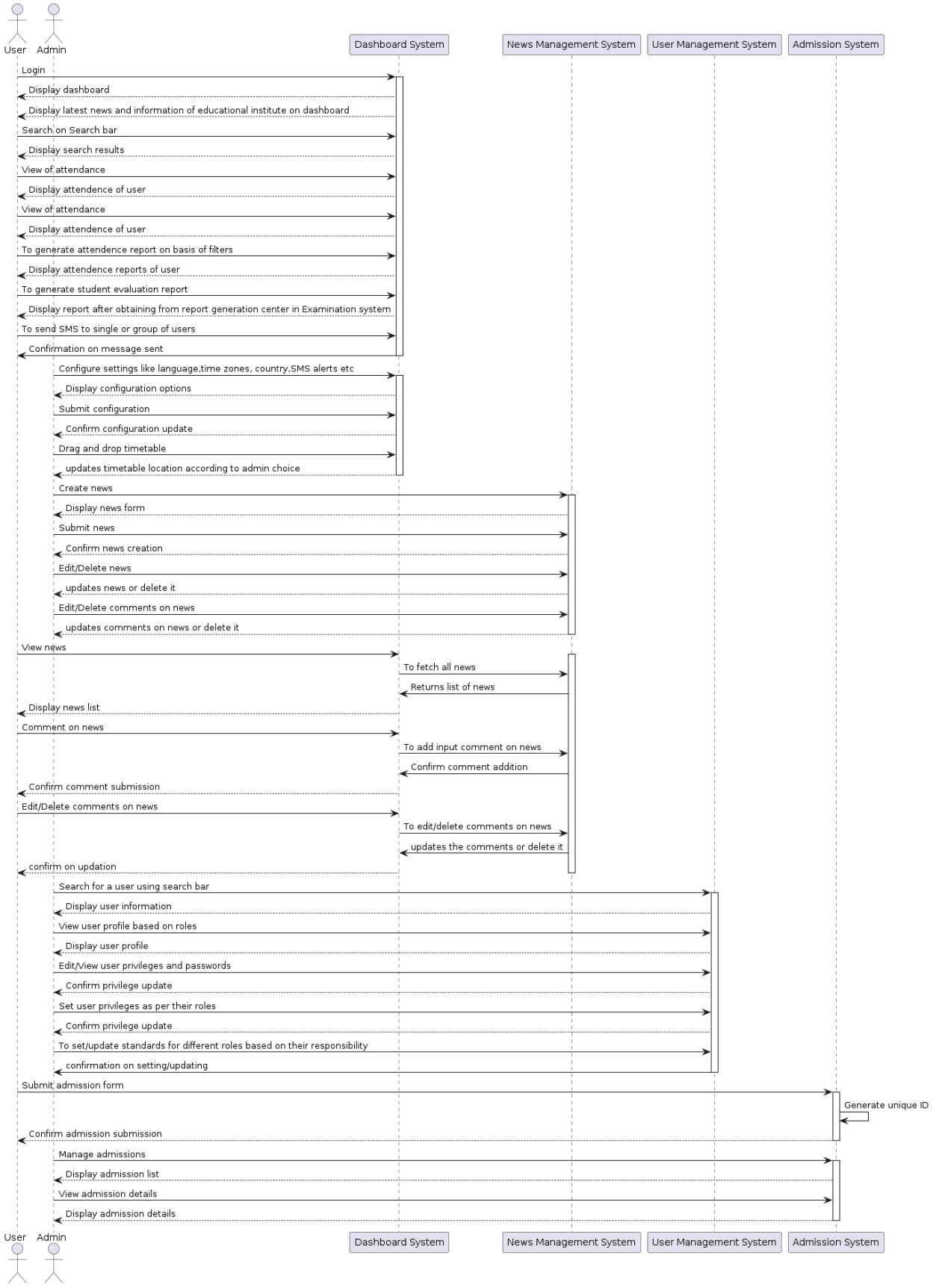
Class Name	Responsibilities
Time_table	Create, edit, delete, and manage timetable entries, generate timetables, alert subject, and class limits, and handle timetable operations like dragging and deleting.
Messaging	Send messages, auto-send messages, and record communication between users.
Image	Store image data for various purposes such as student photos or admission confirmation forms.

Contact	Store contact information including name, phone number, email, and address.
Admission	Create, update, retrieve, and delete admission records for students, manage guardians, emergency contacts, photos, and previous education details.
Guardian	Store information about guardians such as name and relationship
LocalAdmission	Handle local admission processes, such as adding local guardians and uploading admission confirmation forms.
Transfer student	Handle transfer student processes, such as uploading previous college transcripts and adding local guardians.
Comments	Add, delete, and edit comments on news or other content.
Attendance	Mark course and sports attendance, and show attendance records for students.
Student	Add, edit, delete, view, and search student details, including photos, personal information, and academic details
International Admission	Handle international admission processes, such as adding local guardians, uploading admission confirmation forms, and managing visa and passport information.
News	Publish, edit, delete, search, and view news, and manage comments on news.
Fee collection	Set and get fee collection dates and payment options.
Finance	Manage fees, expenses, and incomes, generate reports, approve payslips, and manage assets and liabilities.
FeeAnalyzer	Analyze defaulters and fee structures.
FeeStructure	Set and get fee amounts, due dates, semesters, and academic years.
FeeClassification	Set and get fee classifications.
PayslipApprovalSystem	Approve, add, remove, and get payslips.

Payslip	Set and get basic salary, institution name, allowances, and deductions, calculate total earnings and deductions, and generate payslips.
AssetManager	Add, remove, and get assets.
UserMangement	Add, remove, search users, view user profiles, and edit user passwords.
Role	Set and get role names and privileges.
User	Set and get usernames, passwords, and roles.
basic config	Store basic configuration information such as country, currency, and time zone.
Dashboard	Choose a language, display news, set and edit configurations, send SMS, activate the SMS module, and set SMS configurations.
Privilege	Set and get privilege names and descriptions.
Exam	Calculate grades, GPA, CWA (Continuous Weighted Average), and CCE (Continuous and Comprehensive Evaluation). Manage marks for exams.
exam report:	Generate general, statistical, graphical, and chart reports for exams.
HumanResources:	Approve or deny leave requests. Generate payroll forms. Generate admission forms.
Course:	Add and remove courses. Assign instructors and teaching assistants to courses. Manage enrolled students and teaching assistants for courses.
Batch:	Manage batches of students. Add and remove students from batches.

Pictures of Sequence Diagram


Admin
Student
Teacher
Admission System
Student Management System
Examination System
User Management System
Submit comprehensive admission form
Generate unique ID
Confirm admission
Store the details of student's past education for records
Add/update student guardian details
Guardian details updated/added
to fetch student details in emergency
returns emergency details
to upload/update student photo
Updated the photo of student successfully
Advanced Search facility for existing/former students by specifying filters with inputs
Display corresponding student details
View all students based on batches
Returns list of corresponding students
Create exam based on grades,marks etc
Exam created
Conduct exam
Exam results
Conduct Group exams
Group Exam conducted
Search for a user using search bar
Display user information
View user profile based on roles
Display user profile
Edit/View user privileges and passwords
Confirm privilege update
Set user privileges as per their roles
Confirm privilege update
To set/update standards for different roles based on their responsibility
confirmation on setting/updating
View student profile
Display profile details
To Generate quick ,automated and on demand report
Instructing to use GPA,CCE and CWA methods for student evaluation
Display report statistically and in form of charts for better analytical view
View all reports
Display all reports statistically and graphically
Admin
Student
Teacher
Admission System
Student Management System
Examination System
User Management System


User
Admin
Dashboard System
News Management System
User Management System
Admission System
Login
Display dashboard
Display latest news and information of educational institute on dashboard
Search on Search bar
Display search results
View of attendance
Display attendence of user
View of attendance
Display attendence of user
To generate attendence report on basis of filters
Display attendence reports of user
To generate student evaluation report
Display report after obtaining from report generation center in Examination system
To send SMS to single or group of users
Confirmation on message sent
Configure settings like language,time zones, country,SMS alerts etc
Display configuration options
Submit configuration
Confirm configuration update
Drag and drop timetable
updates timetable location according to admin choice
Create news
Display news form
Submit news
Confirm news creation
Edit/Delete news
updates news or delete it
Edit/Delete comments on news
updates comments on news or delete it
View news
To fetch all news
Returns list of news
Display news list
Comment on news
To add input comment on news
Confirm comment addition
Confirm comment submission
Edit/Delete comments on news
To edit/delete comments on news
updates the comments or delete it
confirm on updation
Search for a user using search bar
Display user information
View user profile based on roles
Display user profile
Edit/View user privileges and passwords
Confirm privilege update
Set user privileges as per their roles
Confirm privilege update
To set/update standards for different roles based on their responsibility
confirmation on setting/updating
Submit admission form
Generate unique ID
Confirm admission submission
Manage admissions
Display admission list
View admission details
Display admission details
User
Admin
Dashboard System
News Management System
User Management System
Admission System


Admin
Employee
Teacher
Student
Parent
HR System
Attendance System
Finance System
Messaging System
Timetable System
Manage employee
Display employee form
Submit employee data
Confirm employee addition/update
Request leave
Approve/reject leave?
Submit leave decision
Notify leave status
Manage entire payroll efficiently
Compute the payroll of employees doing all salary deductions and bonuses etc
Send the payroll data to generate payslips
Automatic processing,calculations and payslip generation
returns payslips with its approval
Display payslip for approval
One-click payslip approval
Confirm payslip approval
Reject payslip
Confirm payslip rejection
Rechecking of payroll data
Tells about payslip rejection by admin
Mark attendance
Display attendance form
Submit attendance
Confirm attendance submission
Can add notes/remarks on attendence given
Manage Assets,Liabilities and Donations
Manage fees
Display fee management options
Set fee structure
Confirm fee structure update
Fetch list of fee defaulters and their details
return the list of fee defaulters
Efficient Fee collection date designing system
Pay fees instantly
Display payment form
Submit payment
Confirm payment by automatic transactions
managing all expenses and incomes efficiently
Different reports generation
return the reports for analysis
To assign tutor to students
Tutor assigned based on batches
Send message
Display message form
Submit message
Receives messages from parents, students and teachers
Prior information regarding school events,news and holidays
Receive messages
Send messages to others
Receive messages
Send messages to others
Receive messages
Send messages to others
Create timetable
Display timetable form
Submit timetable
Alerts on timetable creation
Edit/Delete timetable
Returns updated timetable
No alerts
Confirm timetable creation
Edit/Delete timetable
returns updated timetable
View timetable
Display timetable

The design reflects a balance among competing criteria as :

- **Separation of Concerns**: We have incorporated this issue by dividing the system into distinct classes, each responsible for a specific set of related functionalities. For example, the Time_table class handles all timetable-related operations, while the Admission class manages admission processes. This separation promotes solving a concern very easily by dividing the concern into smaller concerns, each one dealing with one or more classes and then solving them independently to solve a big concern. The presence of multiple classes helps in separation of concerns.

- **High Cohesion :** High cohesion refers to the degree to which the elements within a module or component are related to each other. In other words, high cohesion means that the responsibilities of a module or component are closely related and focused on a single task or purpose. Modules with high cohesion are easier to understand, maintain, and reuse because they have clear and well-defined responsibilities. In our design, we have made several classes of each module in order to understand each module easily and support high cohesion and each class has its own functionality. For example in our design News Management system class is very functional and deals only with News and comments and can be easily understandable and can be taken out and used in some other design easily.

- **Low Coupling**: Low coupling refers to the degree of interdependence between modules or components within a system. It measures how much one module relies on other modules to accomplish its tasks. Low coupling means that modules are relatively independent of each other, and changes to one module have minimal impact on other modules. This improves the modularity and flexibility of the system, making it easier to modify, extend, and maintain. We have accomplished low coupling through the use of interfaces or abstract classes, which can decouple the implementation details from the client classes. For instance, the FeeAnalyzer class depends on the FeeStructure class, but it interacts with the FeeStructure interface rather than being tightly coupled to a specific implementation.

- **Information Hiding**: We have done the information hiding by encapsulating the internal state and implementing details of classes within their respective classes. We have achieved it using a ‘private’ access modifier, which can only be accessed through well-defined public methods and not of other classes. For example, the Payslip class encapsulates the details of calculating salaries, allowances, and deductions, exposing only the necessary methods for setting and getting relevant information. Also only admin is having powers to access all management systems which resulted in controlled interfaces for users.

- **Law of Demeter**: The classes adhere to the Law of Demeter by limiting direct interactions with other classes. For instance, the Dashboard class interacts with other modules through well-defined interfaces rather than directly accessing their internal components.
- **Extensibility and Reusability:** The design allows for easy extension and reuse of components. For example, the Course class can be easily extended to accommodate additional functionalities related to courses. Similarly, the user management class provides a framework for managing users with different roles, facilitating reusability across different parts of the system. Also many systems like news management and finance management can be reused in other designs easily and can be extensible also.
- **Design Patterns**: We have not explicitly used any of the patterns in the diagram but some of the patterns can be inferred from the diagram. For instance, the use of a centralized Dashboard class for coordinating interactions with various modules hints at the Facade pattern, providing a simplified interface to a complex system. Additionally, the composition of classes and their relationships suggests the use of the Composite pattern, allowing complex structures to be treated uniformly. Another pattern is Information Expert of GRASP in which some classes can retrieve information of others like news management can have comments on news easily.

Strongest Aspects

These are two of the strongest aspects of the design of the system:

- **Encapsulation:** Each class likely holds its own attributes and methods, promoting data protection and modularity. This makes the code more reusable and less prone to errors.
- **Separation of Concerns**: Clear and Organized Class Structure has been used in the class diagram. The class diagram uses clear naming conventions and separates the functionalities into well-defined classes. This makes the code easier to understand and maintain. Also if a concern arises then it will be solved easily.

Weakest Aspects:

These are two of the main weakest aspects of the design of the system:

- **Missing Methods and Details**: While the class structure is clear, some details about methods and attributes are missing as it is not possible to show all the methods with descriptions in the classes. The constructors and destructors of the classes are not mentioned to keep the diagram clear. Understanding the specific functionalities would require further analysis of the code itself.
- **Potential Redundancy**: There might be slight redundancy in some classes and how some functionalities are handled across different modules. Some data is stored in multiple classes which introduces redundancy in the code. For example, User Management and Employee Management might have overlapping functionalities.

Assumptions:

1) In the sequence diagrams we have made an assumption that any message from one actor or instance to other will have two meanings- one is it may be normal message, order or instruction to perform any method/operation or it might be some transfer of data or information like payroll,payslip,details etc.
2) In class diagram, in one class we have assumed circle to equivalent to ‘-’ i.e private access modifier in front of attributes and box or square is equivalent to ‘+’ i.e public modifier and it is because of UML Tool only.