

SSAD Asssignment-4

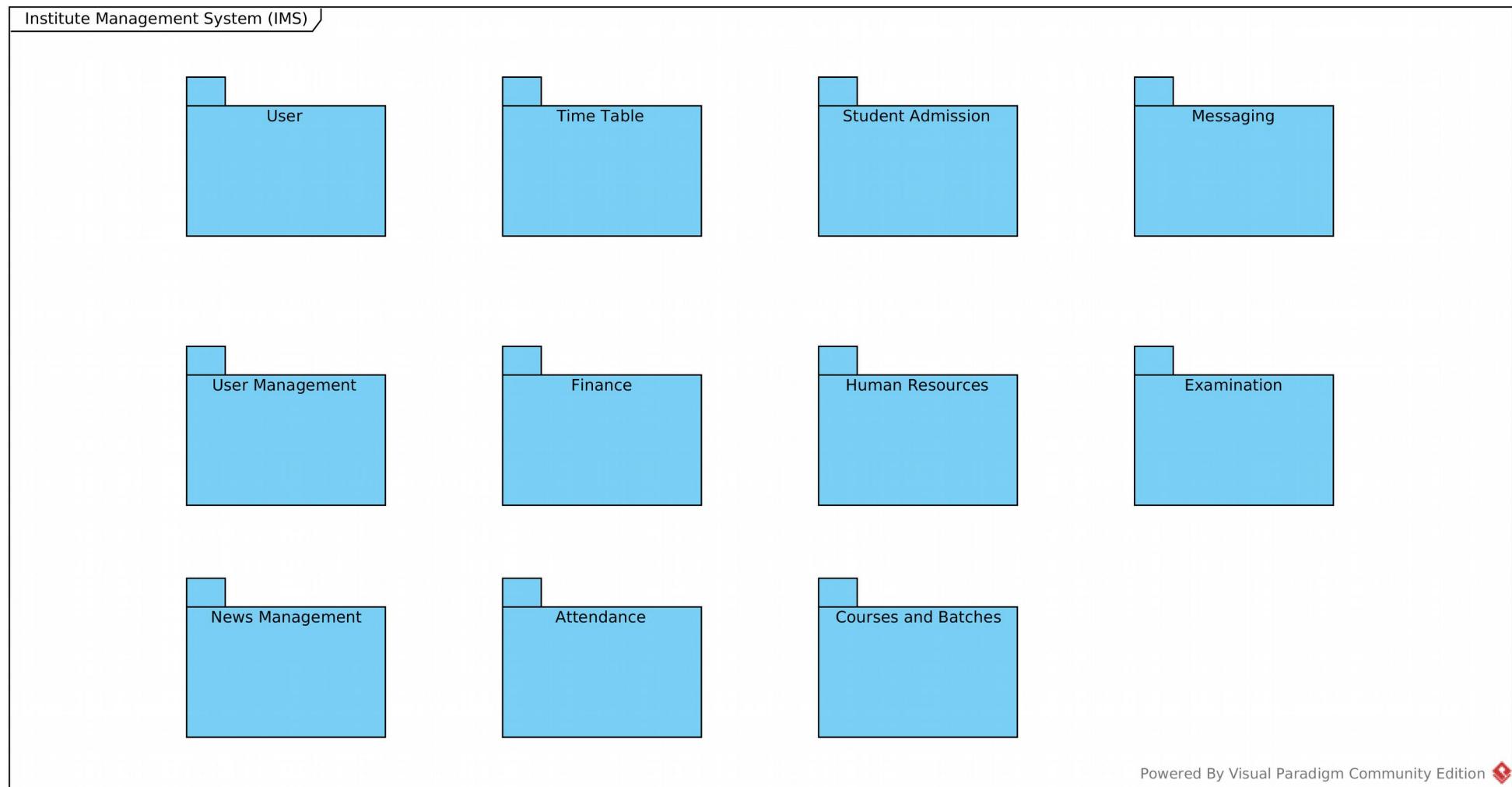
Team members:

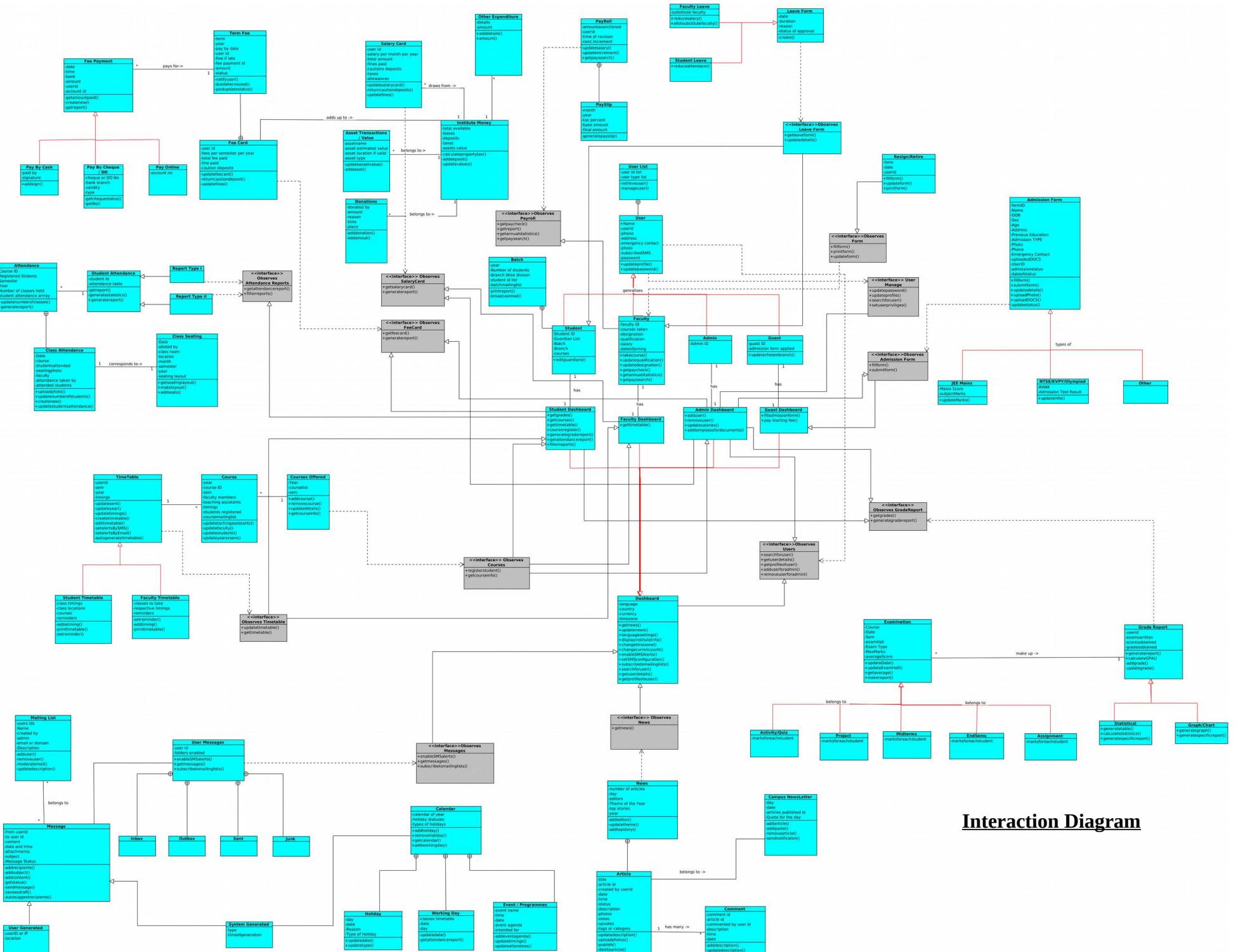
Sai Teja Reddy Moolamalla (201564086)
Hemanth Vemuri (201564126)

1. Tool name used for modeling design diagrams: Visual Paradigm

2. UML Class Diagrams

Package Level:

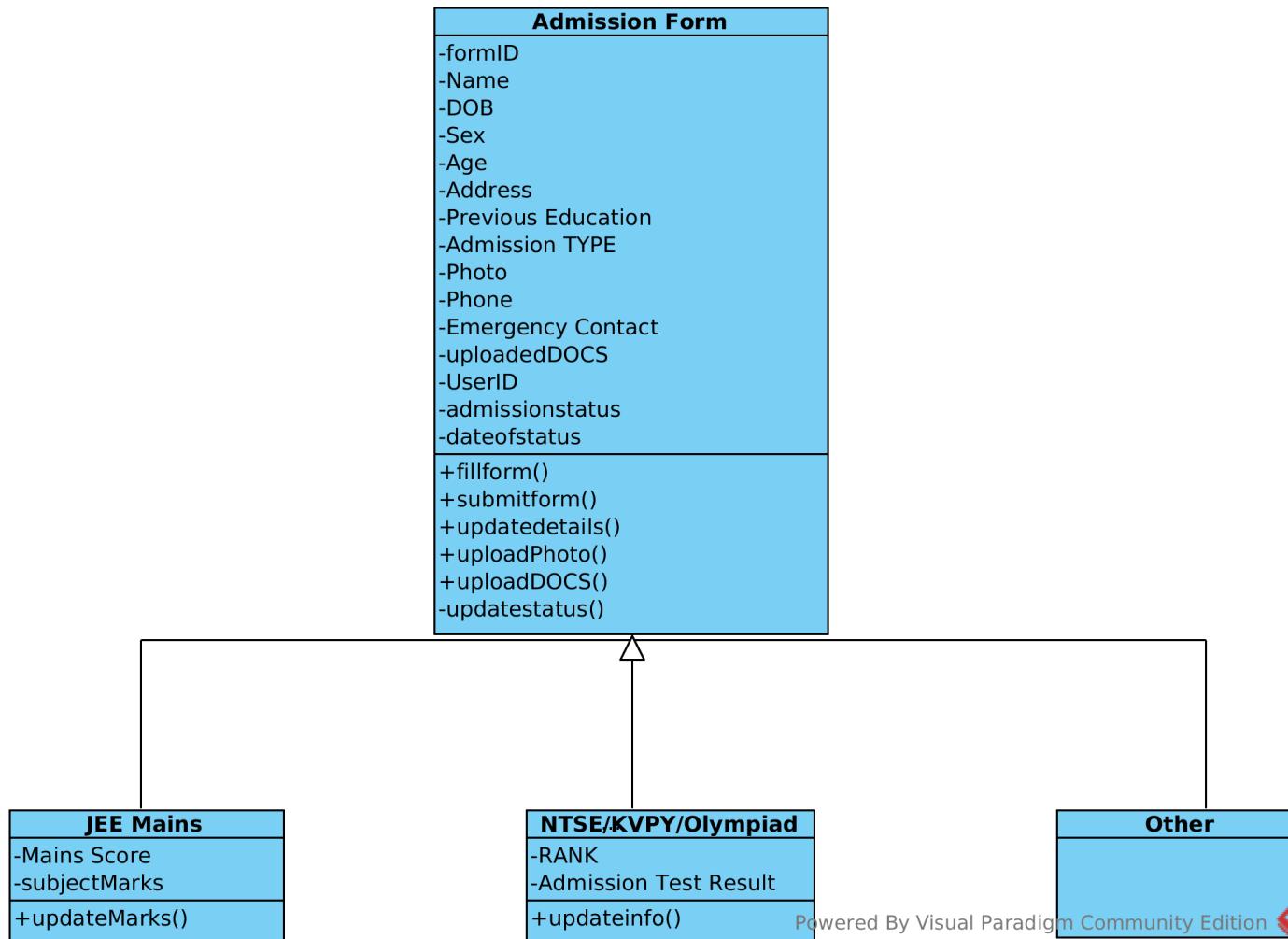




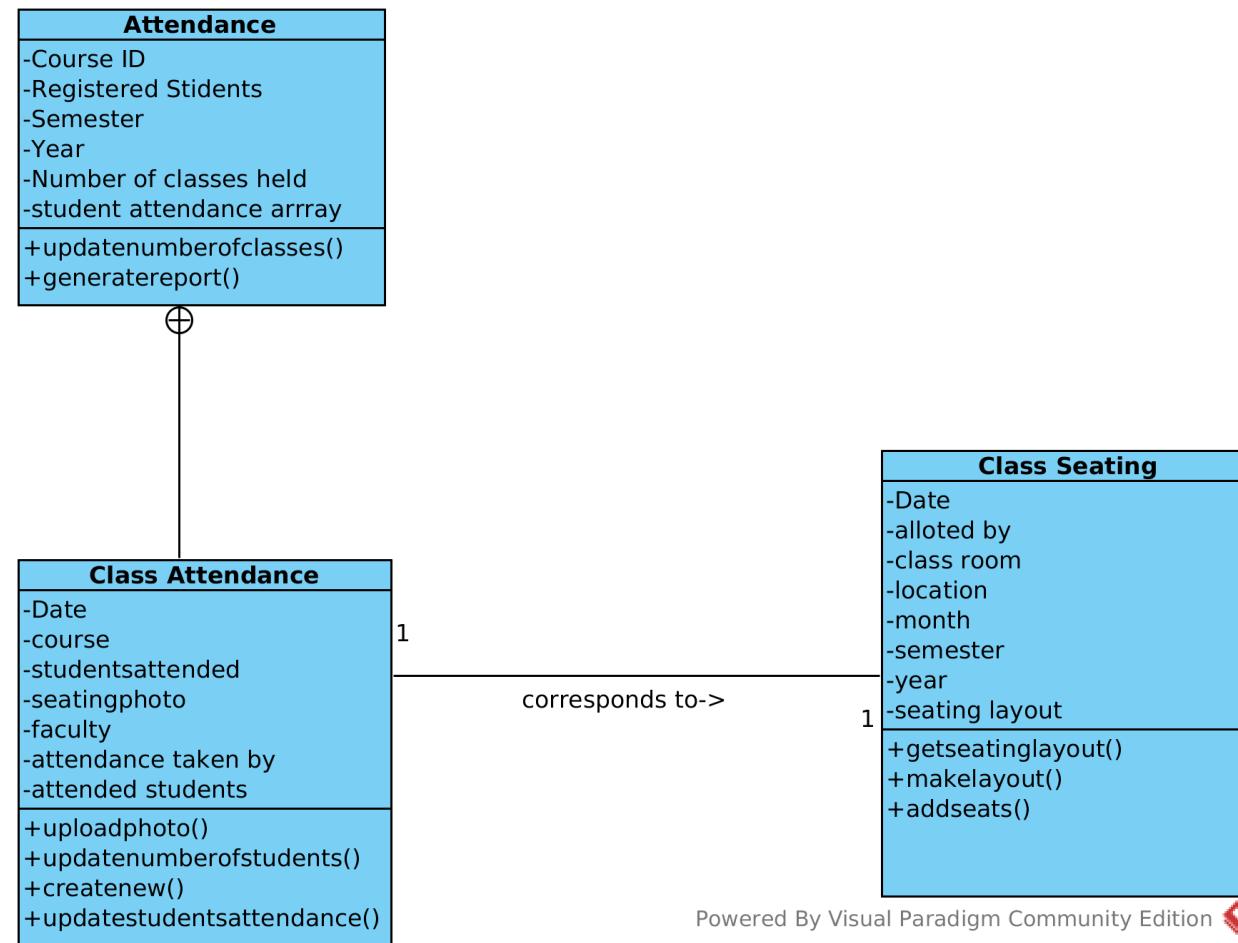
Interaction Diagram

Module Levels:

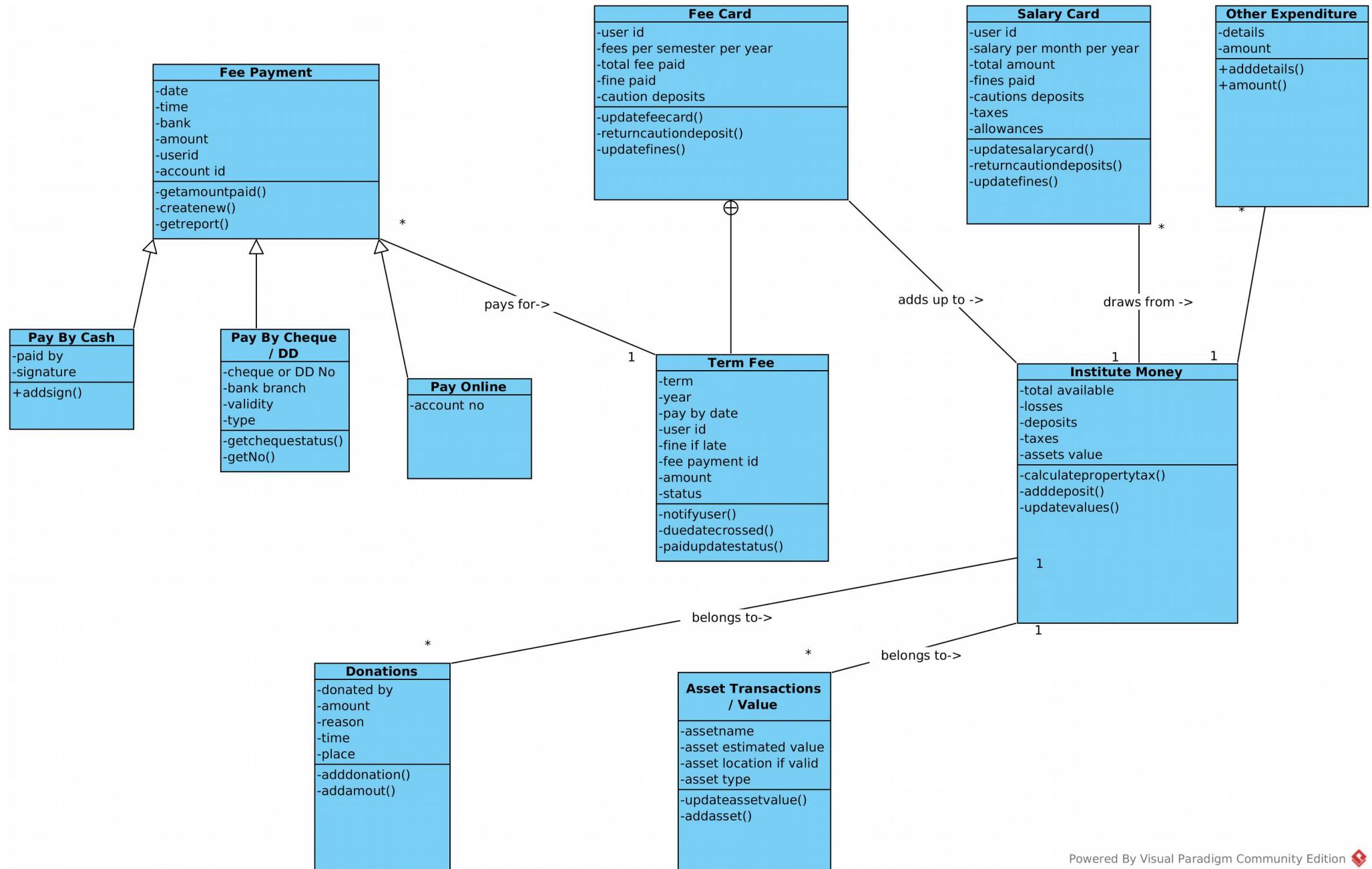
1. Admission Module:



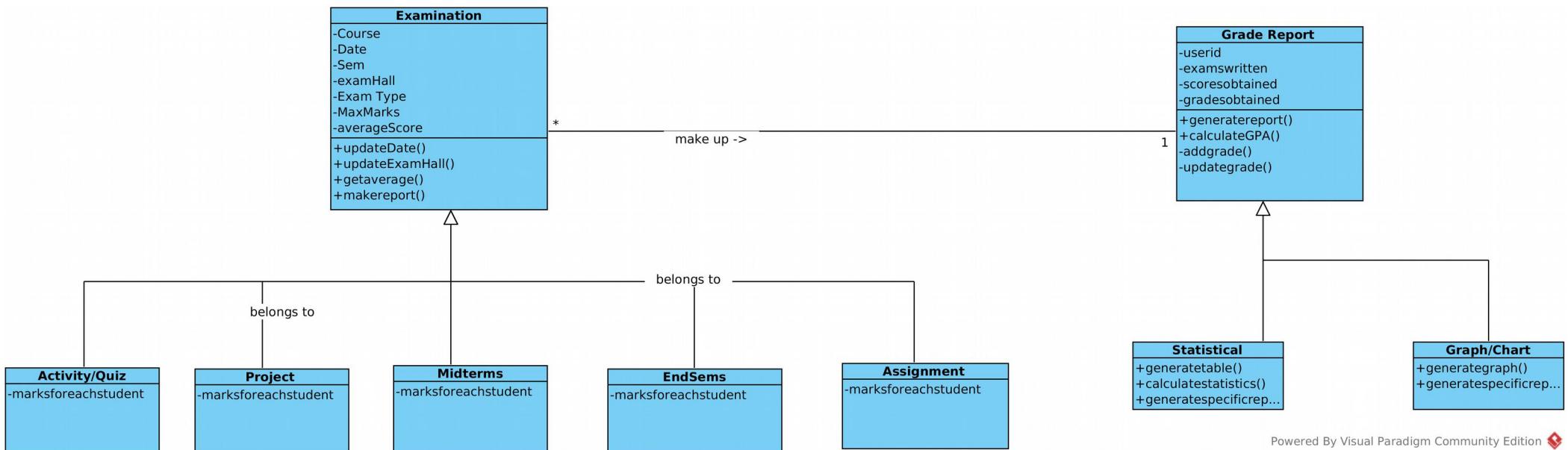
2. Attendance Module:

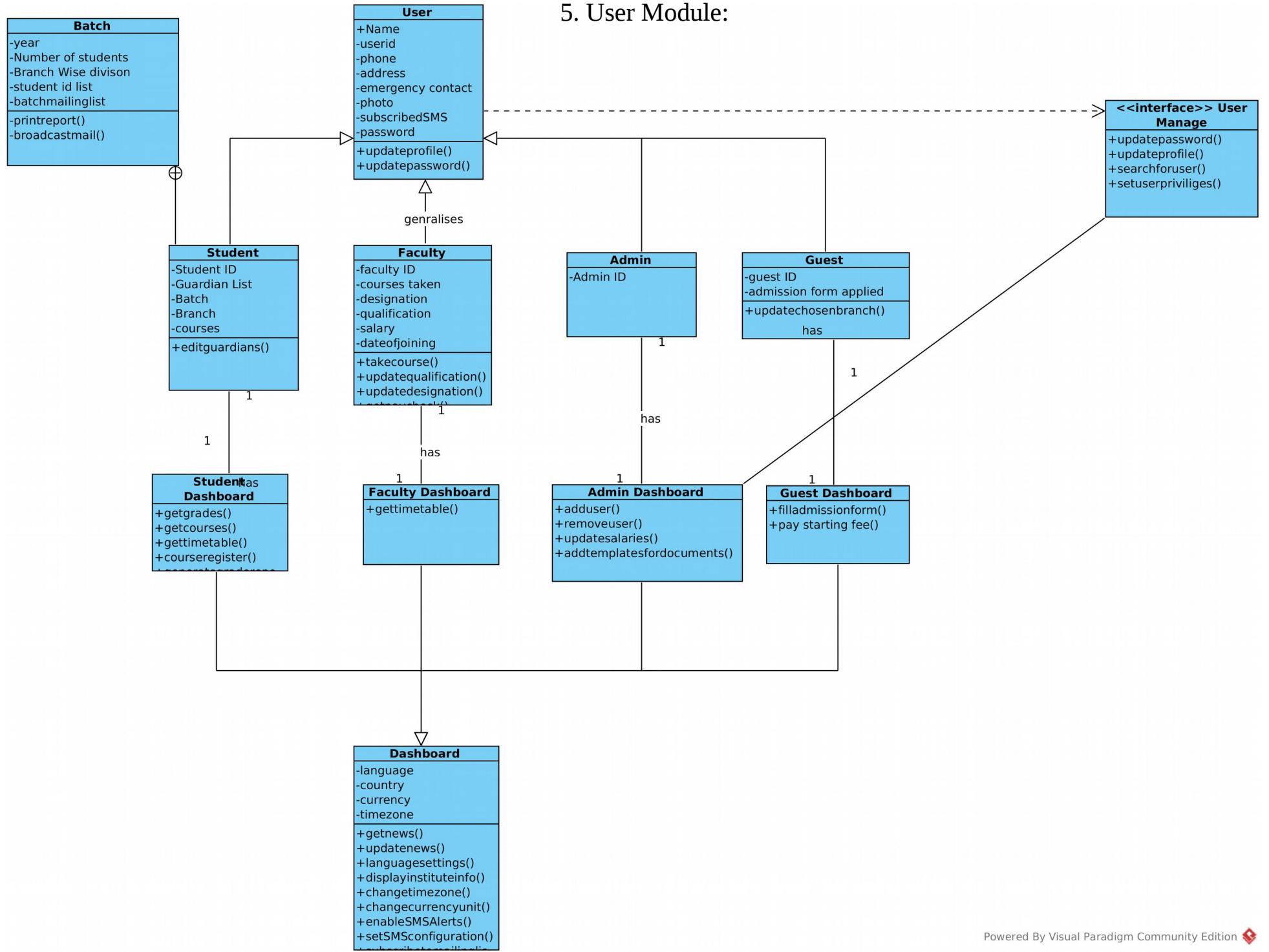


3. Finance Module:

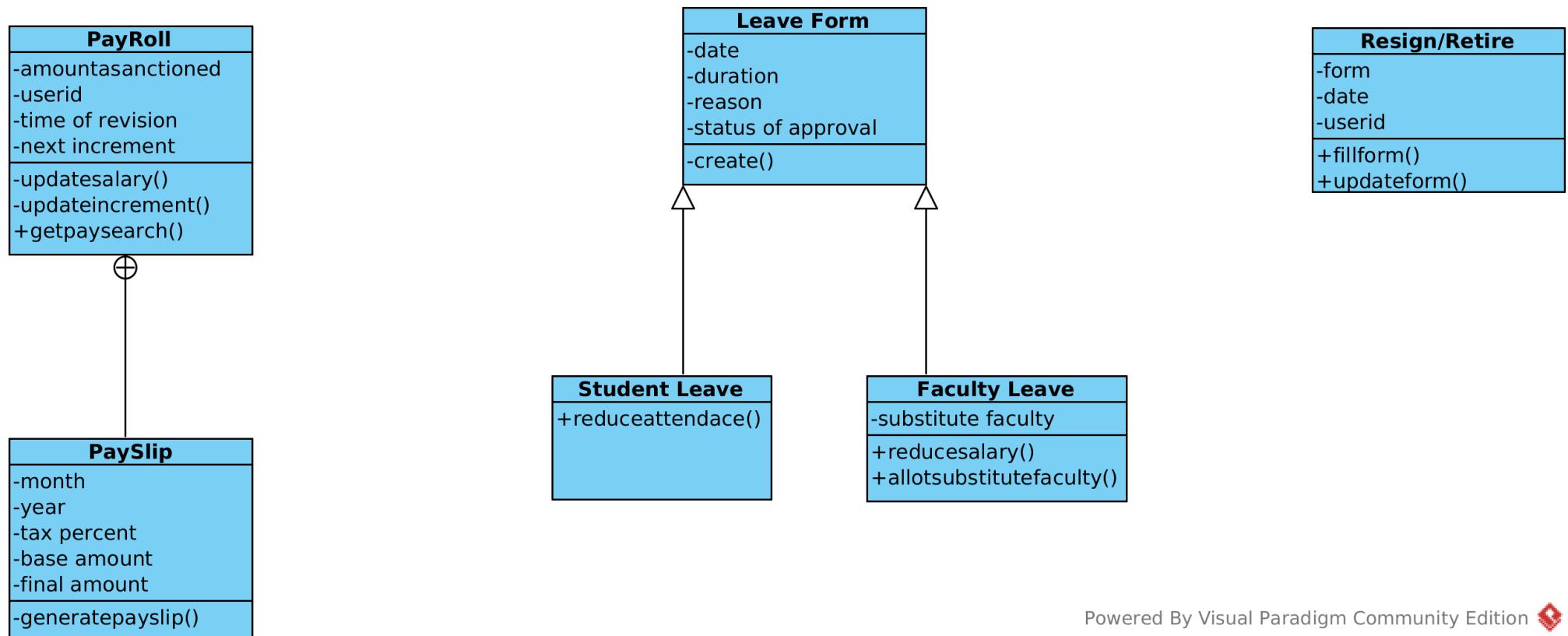


4. Academics Module:

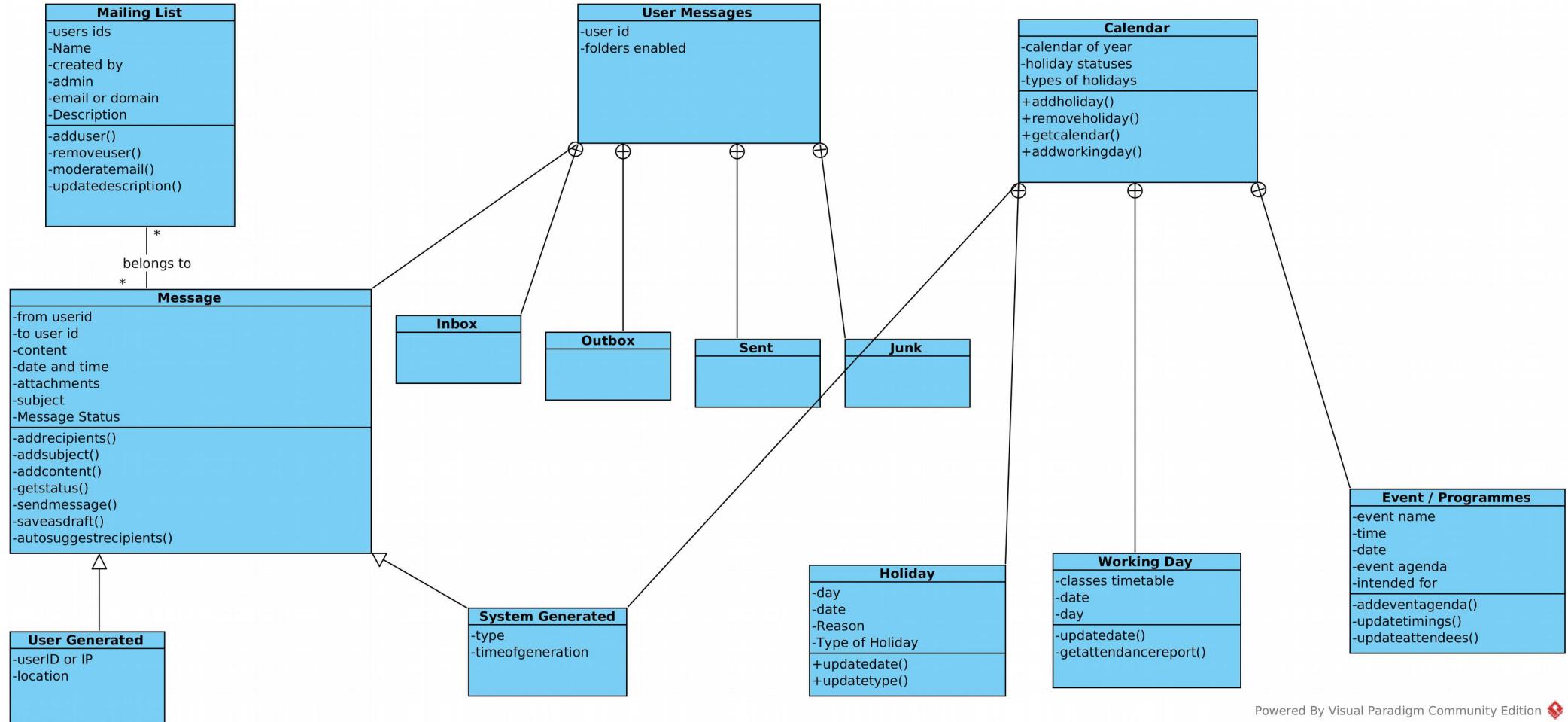




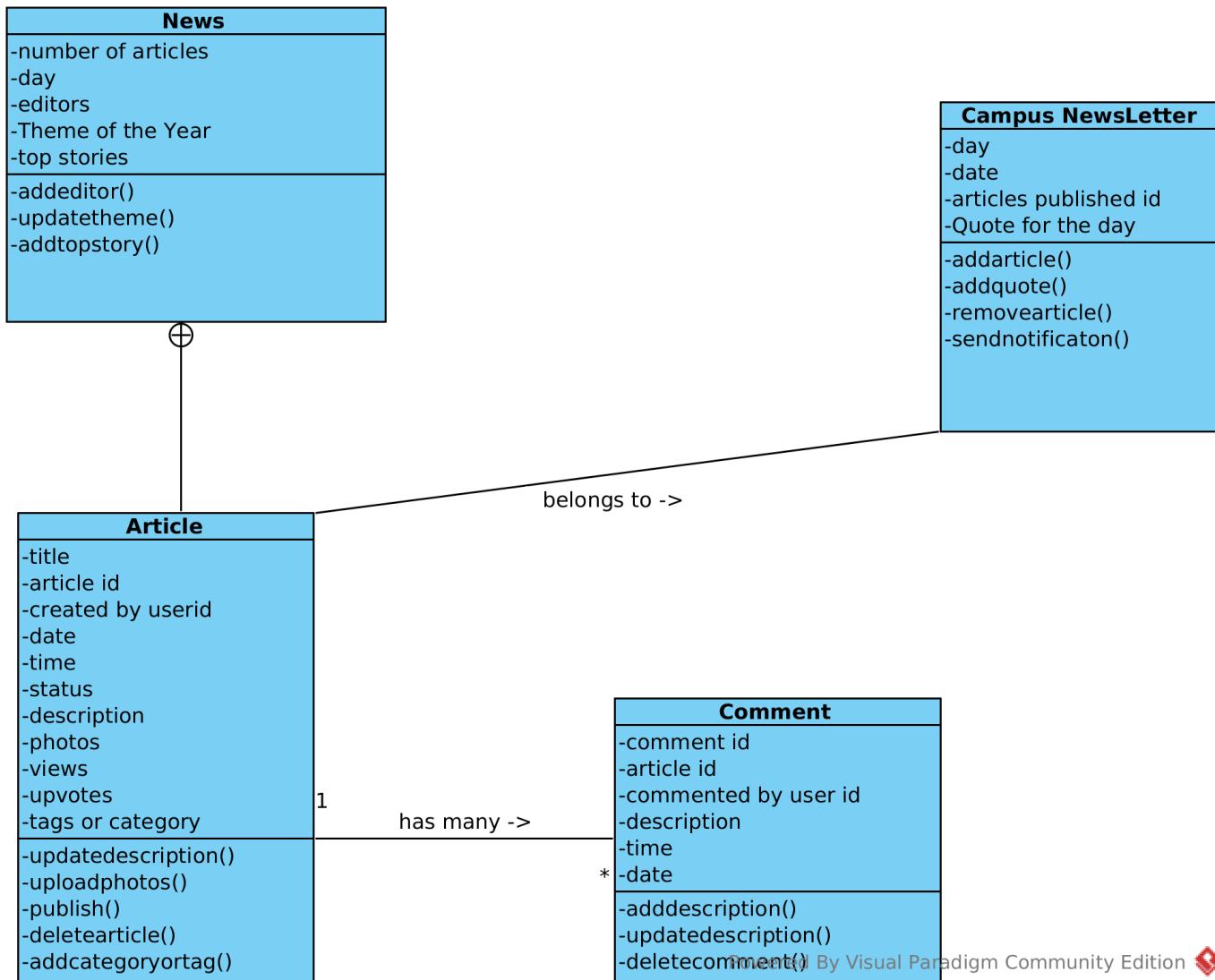
6. Human Resource Module:



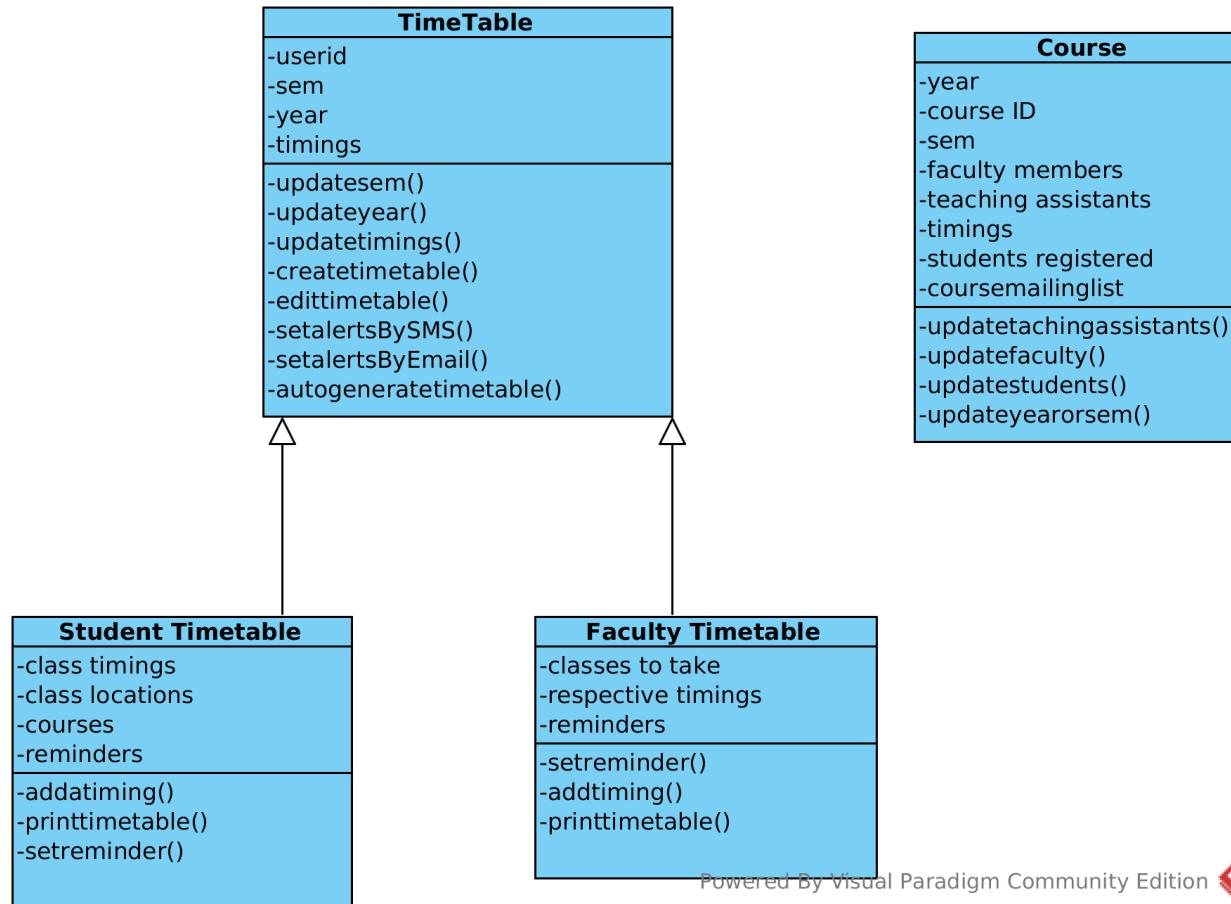
7. Messaging module:



8. News Management Module:



9. TimeTable Module:



3. Table summarizing the responsibilities of each major class.

S.no	Class name	Package	Responsibilites
1	Examination	Academics	Contains the information of an examination like type,examhall,semester and course with which it is associated to along with methods to update this information. It is specialized into Activity/quiz, project, midterms, endsems and Assignments.
2	Grade Report	Academics	Contains a foreign key to the user it is associated with along with exam_id, scoreobtained and grade with methods to calculate the GPA, generate-report and updategrade. Its is specialized to statistical and graphical grade report.
3	Institute Money	Finance	Contains information about the monetary status of the institute, with attributes like assets value, losses, depoists and methods like calculatepropertytax, add-deposit.
4	Salary Card	Finance	Contains the information of the salary of the employees it is associated to using user_id, with attributes like salary per month, caution deposit, taxes, allowances and methods like updatesalray() , returncautiondeposit.
5	Fee Card	Finance	Contains information of the fee structure of the student it is associated to with attributes like total fee paid, caution deposit and methods like update feecard.
6	Term Fee	Finance	Contains information about fee structure for a particular Term, it is aggregated to Fee card class. Contains attributes like term,year ,user-id, fine etc.with methods like notifyuser, updatetestatus
7	Fee payment	Finance	Contains information about fee payment process with attributes like bank, timestamp, user-id, account-id and methods like getamountpaid, getreport. It is speicalized into pay by cash, pay by cheque/DD and pay Online classes

8	Pay by cash	Finance	Contains extra attributes like signature, paid by and method addsign
9	Pay by cheque/DD	Finance	Contains extra attributes like cheque/DD number, branch and methods like get-cheque-status
10	Pay online	Finance	Contains extra information like account no, payment-portal
11	Donations	Finance	Contains information about the Donations received by the institue with attributes like donatedby, amount, reason, and methods like adddonation and addamount
12	Asset Transaction/ Value	Finance	Contains information about the various assets and their values owned by the institute wiht attributes like name, value, type and methods like update-asset-value
13	Other Expenditure	Finance	Contains information about other expenditure of the institute
14	Admission Form	Admission	Contains information about the admission for admitting new students with attributes contains basic information of the student inaddition to attributes like type of addmision , status, photo , documents with methods like uploadphoto, uploaddocuments updatestatus etc. It is speicalized into JEE mains , NTSE/KVPY depending upon the type of admission
15	Attendence	Attendence	Containa information about the attendence for a course with attributes like courseID , sem, year, number of students, student attendance array with methods like update-number-of-classes, generatereport
16	Class Attendence	Attendence	It is aggregated to the attendence class and contains information about the attendence for a particular date with attributes like seating photo , studenst attended list and methods like uploadphoto, update-sudent-attendence.
17	Class seating	Attendence	Contains information about the seating arrangment of particular class room with attributes like date, location, month , semester, year, and methods like getseatinglayout, addseats

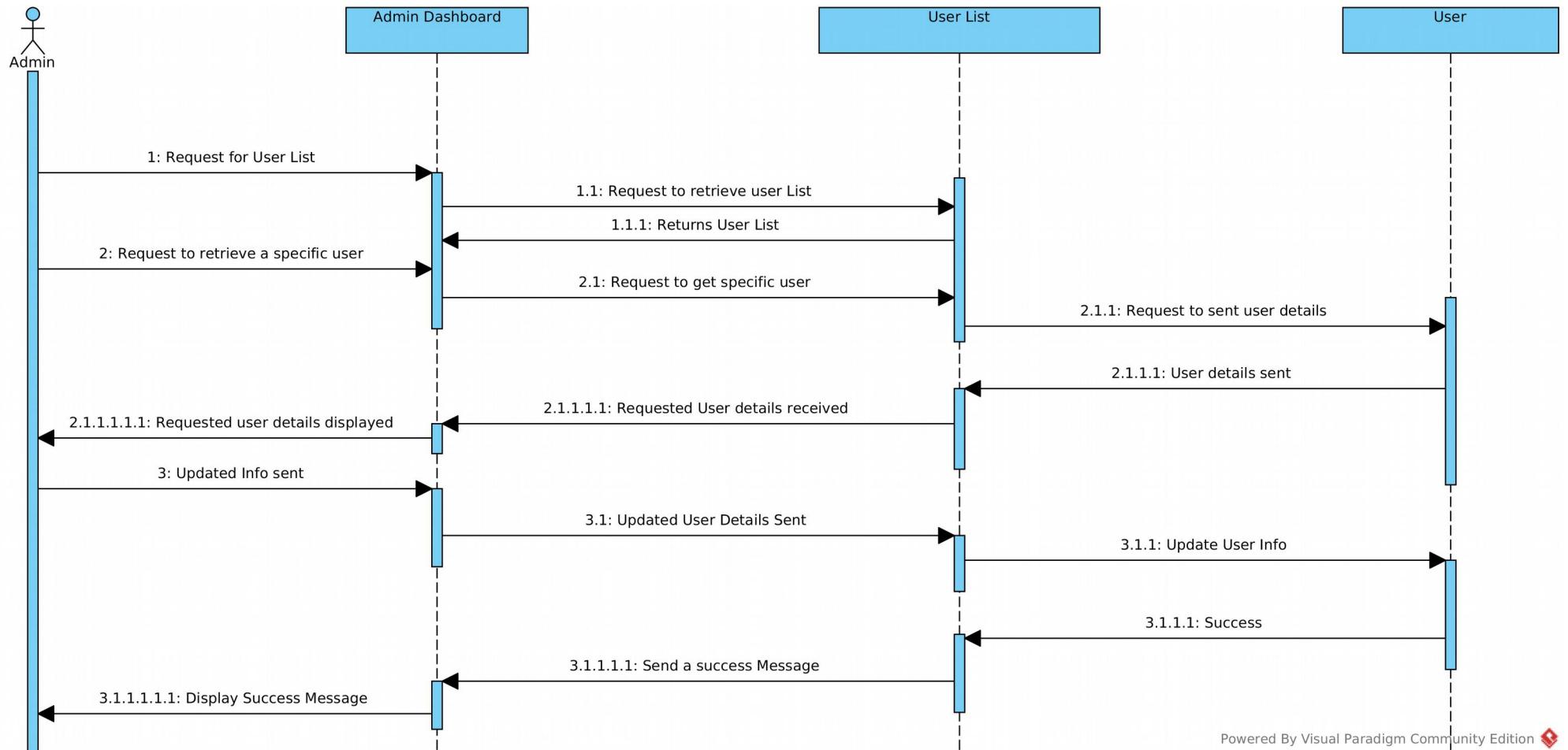
18	Payslip	Human Resource	Contains information about the salary paid to a particular employee with attributes like tax percent, month , year and methods like generatepayslip . It is aggregated to the payroll class
19	PayRoll	Human Resource	Contains information about the overall monetary transactions with an employee with attributes like userid, amount sanctioned and methods like updatesalary, increment
20	Leave form	Human Resource	Contains the information about the leave application process with attributes like date, duration , reason and status with methods like updatetestatus. It is specialized into student leave and faculty leave classes
21	Calendar	Messaging	Contains the information about the almanc of a particular year with attributes like holidays and methods like addholidays, removeholidays . Holidays,working days and event/programmes classes are aggregated to calendar class
22	Mailing list	Messaging	Contains the information about the users in a particular mailing list with attirbutes like user_id list, name , created by, admin, description and methods like edituser, moderate-email, updatedescription.
23	Message	Messaging	Contains the information of a particular message/mail with attributes like from_id, to_id , content, attachments etc and methods like addsubject, addcontent, addfile, savetodraft, etc
24	User message	Messaging	Contains the information about all messages of a particular user with attributes like userID, folders etc. Inbox, Outbox, Sent, Trash classes are aggregated to this class, each of them contains a list of messages
25	News	News	Contains information about the articles related to a particular News with attributes like numberofarticles, year, editors, topstories and methods like addeditor updatetheme etc
26	Article	News	Contains information about news threads with attributes like author, time, date,

			description, photos, views, category and methods publish, updatedescription , addcategory . It aggregates to the new class
27	Comment	News	Contains information about the comments associated with a particular article with attributes like commentID and articleID ,description etc and methods like adddescription , deletecomment, updatedescription etc
28	Campus Newsletter	News	Contains articles about news on a particular day with attributes like day, date, etc and methods like addarticle, sendnotification etc
29	Timetable	TimeTable	Contains schedule for a particular user identified with userID .It contains attributes like sem, year, timmings and methods like updateSem, updateTimings, SendalertbySMS , sendalertbyEmail, auto-generate-timetable . It is specialized into student timetable and faculty timetable
30	Course	TimeTable	Contains information about course identified with a courseId . It contains attributes like faculty, teaching assistants , students registered , mailing list and methods like updatefaculty, updateTA etc
31	User	User	Contains the general information about an user with attributes like name, userid, phone number, address , photo, emergency contact etc and methods like updateprofile, updatepassword.
32	Batch	User	Contains the information of the students in a batch with attributes like year, number of students , student ID with methods like printreport and broadcastmail. The student class is aggregated to student
33	Student	User	Contains additional information like guardian list , bracnh, courses, and methods like edit-guardians
34	Faculty	User	Contains additional information like designation , salary, dateofjoining , courses taken and methods like updatequalificaton, takecourse

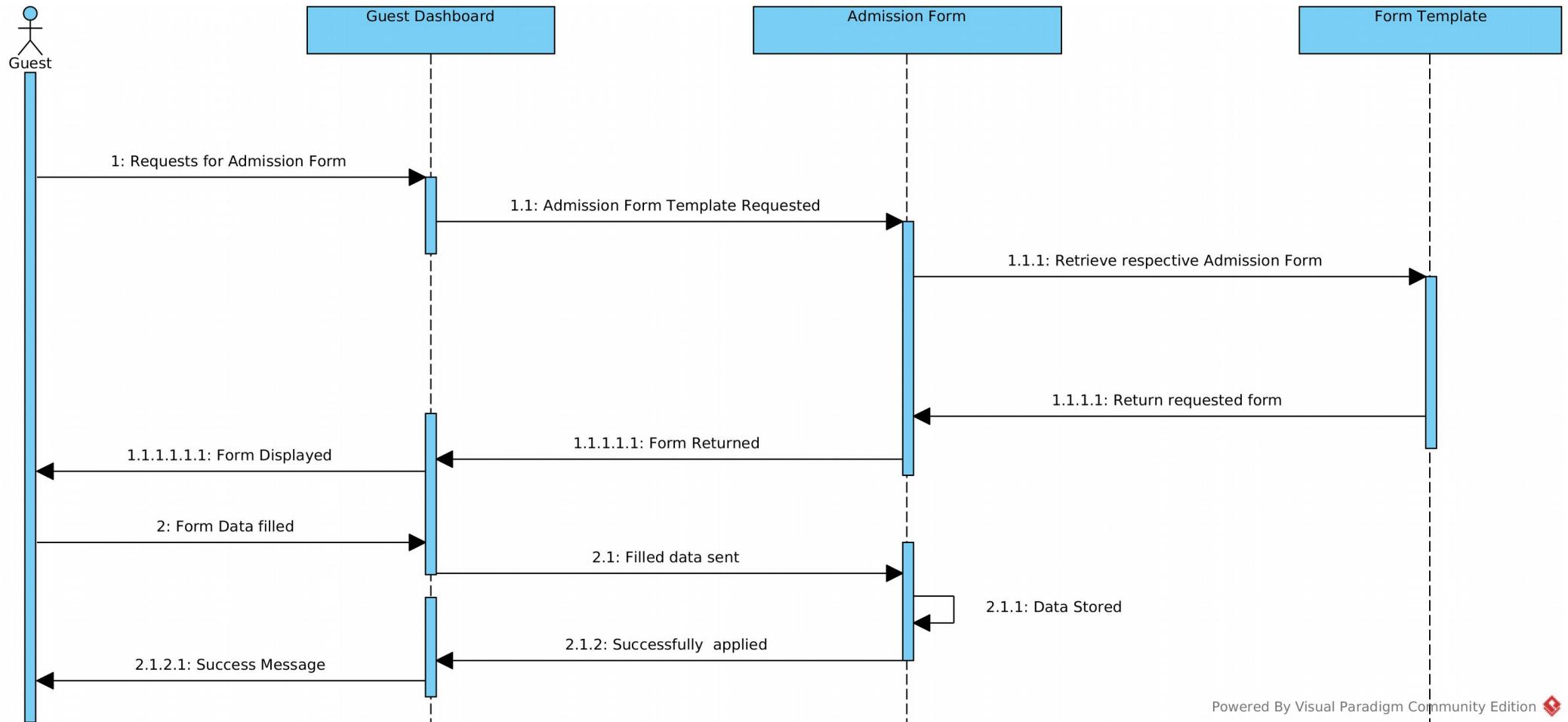
35	Admin	User	Contains extra information pertaining to the admin
36	Guest	User	Contains information of any guest user currently using the system , has attributes like guestID , admission applied and methods like updatechosenbranch
37	Dashboard	User	Contain information present in the dashboard like the time , language, country , and methods like getnews, updatenews, changetimezone, changecurrency , enableSMSalert, setSMSconfiguration. This class is a generalisation of the studentDashboard, facultyDashboard, AdminDashboard and guestDashboard
38	Student Dashboard	User	Contains extra methods like getgrades, getcourses , gettimetable.
39	Faculty Dashboard	User	Contains extra methods required only for the faculty like gettimetable
40	Admin Dashboard	User	Contains methods like adduser, removeuser, updatesalaries
41	Guest dashboard	User	Contains methods like filladmissionform, payfees etc.

4.Sequence Diagrams:

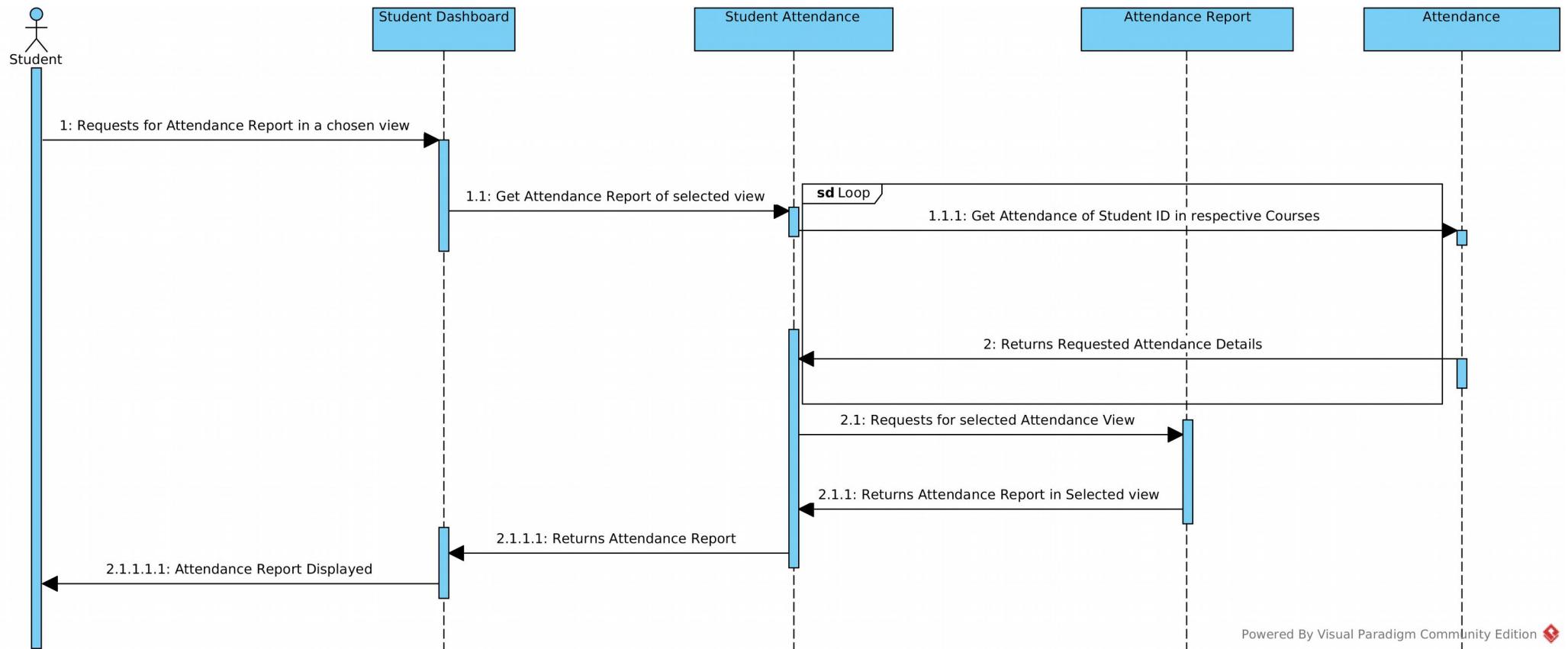
1. Sequence Diagram for the Admin to manage users:



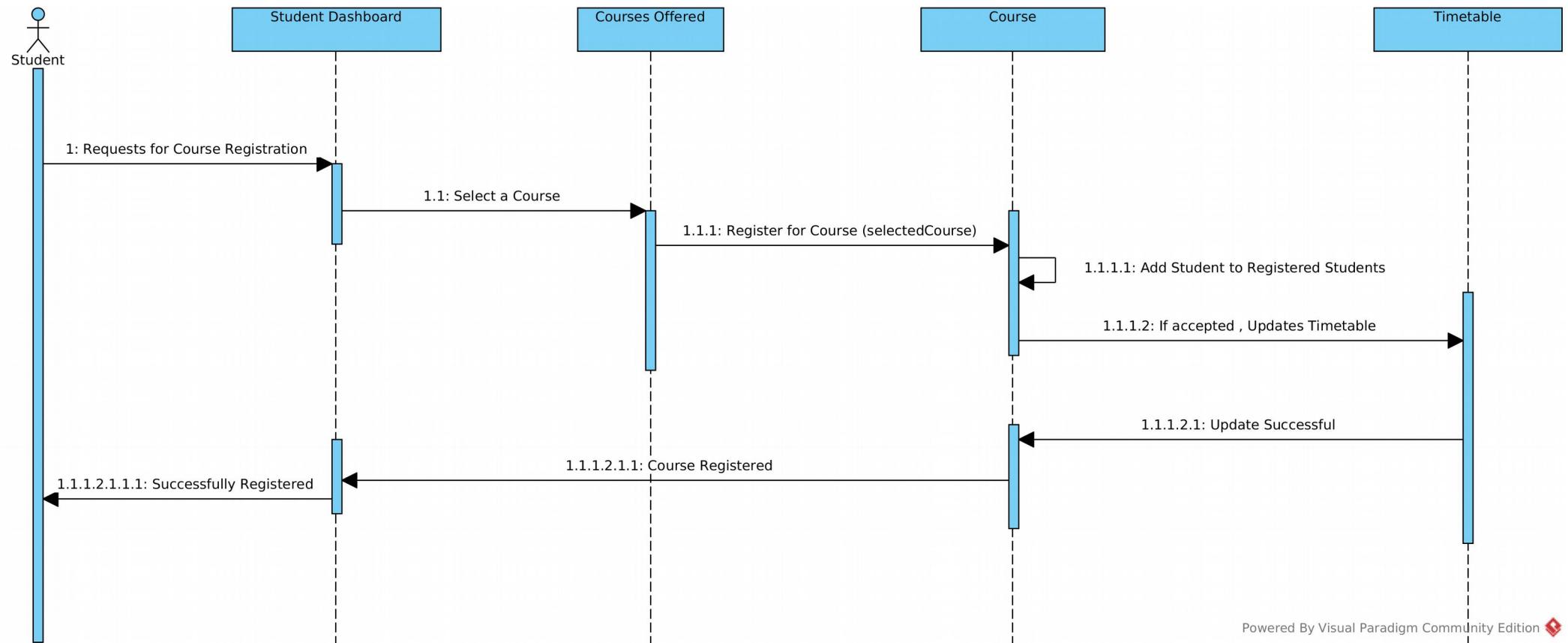
2. Sequence Diagram for Successfull submission of Admission form:



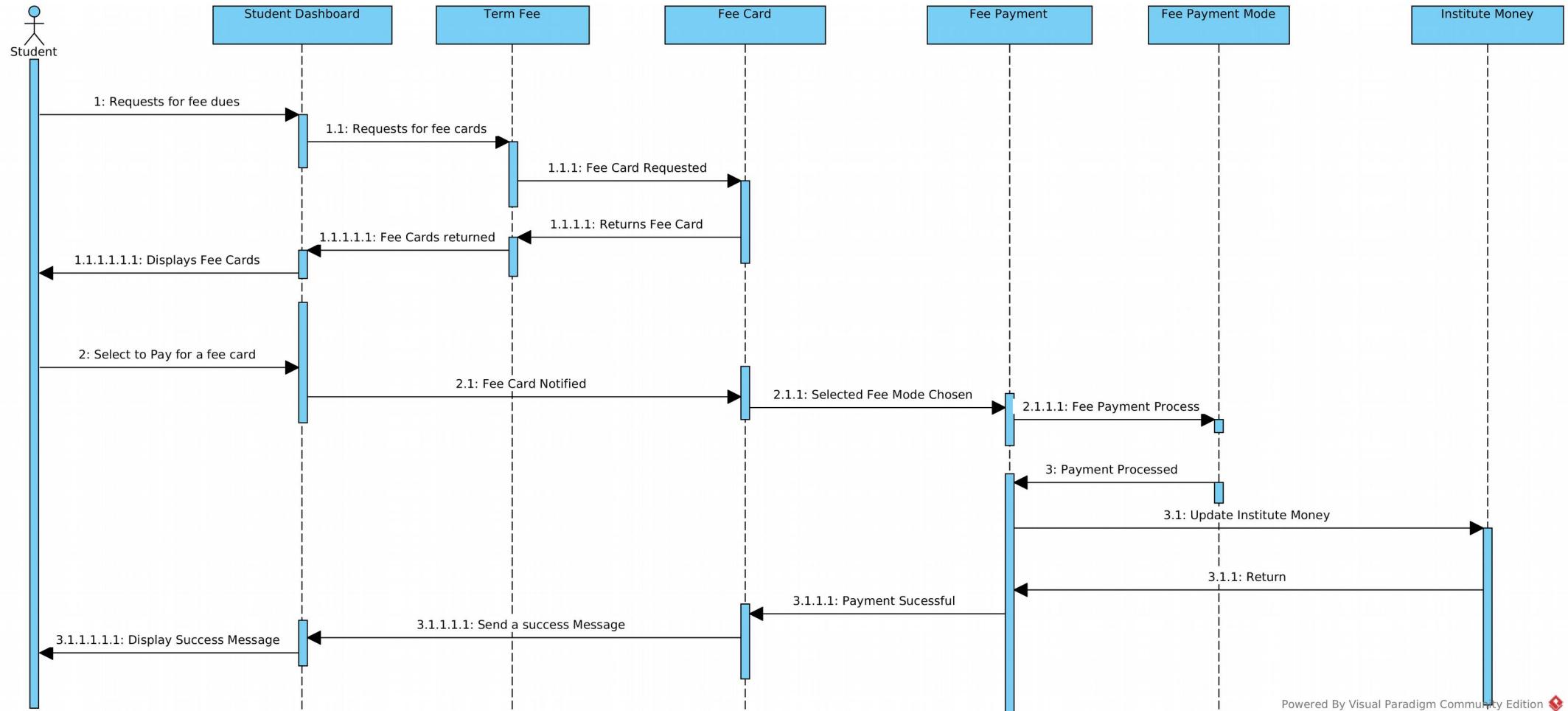
3. Sequence Diagram for Student request for Attendance report:



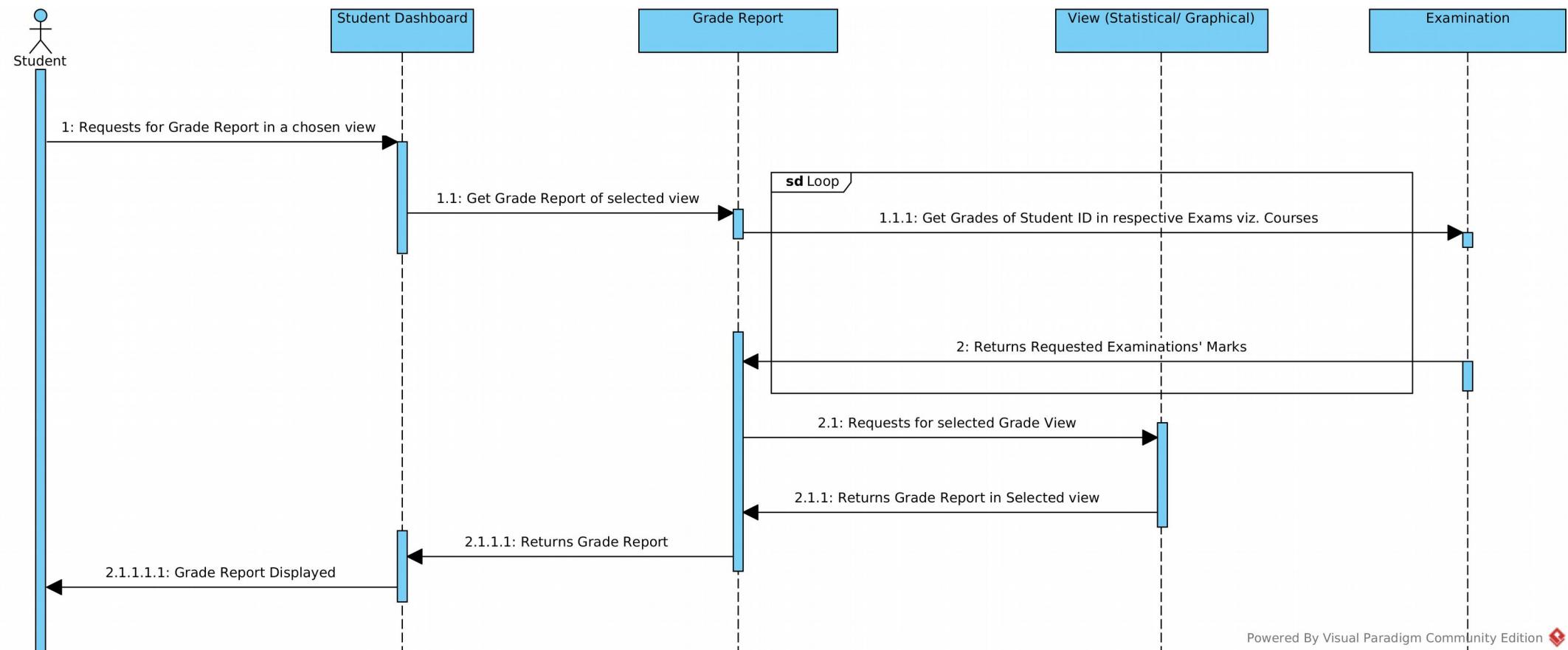
4. Sequence Diagram for New course Registration:



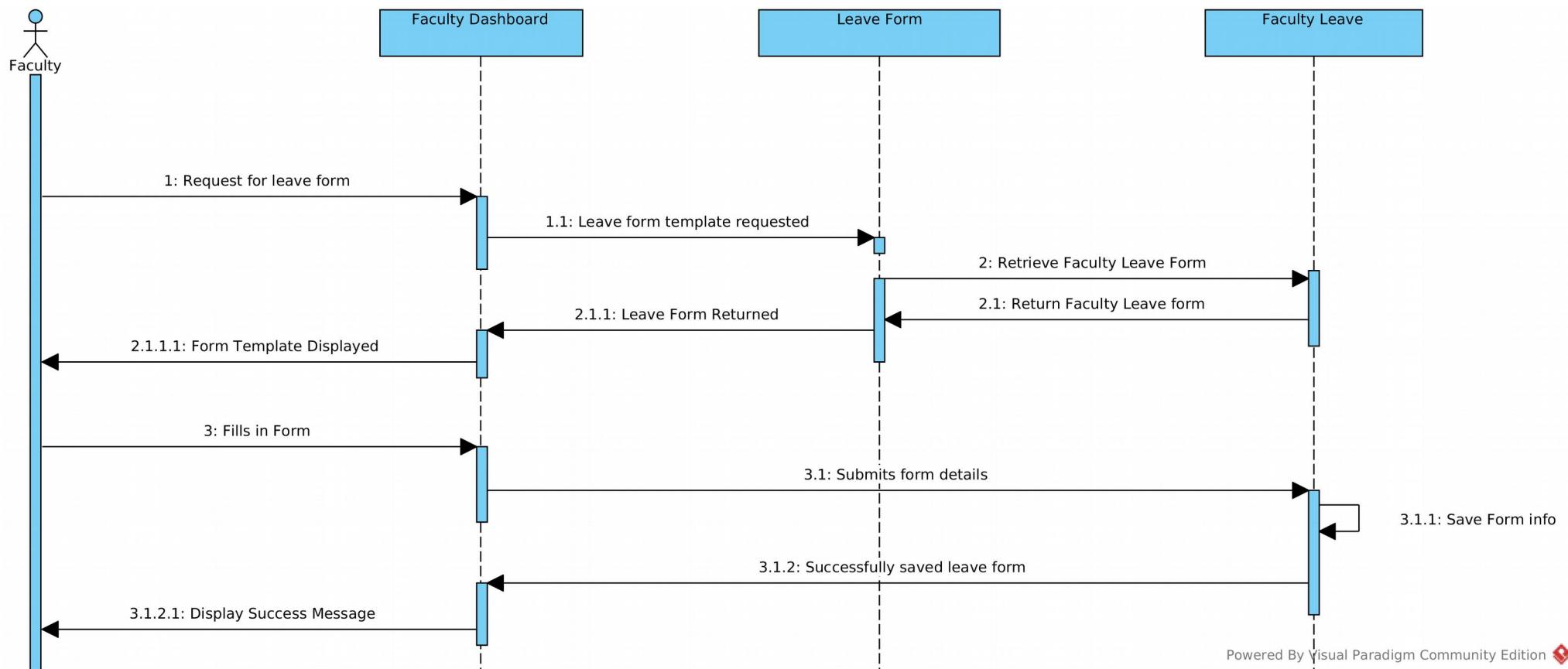
5. Sequence Diagram for Request for fee dues and payment of fees:



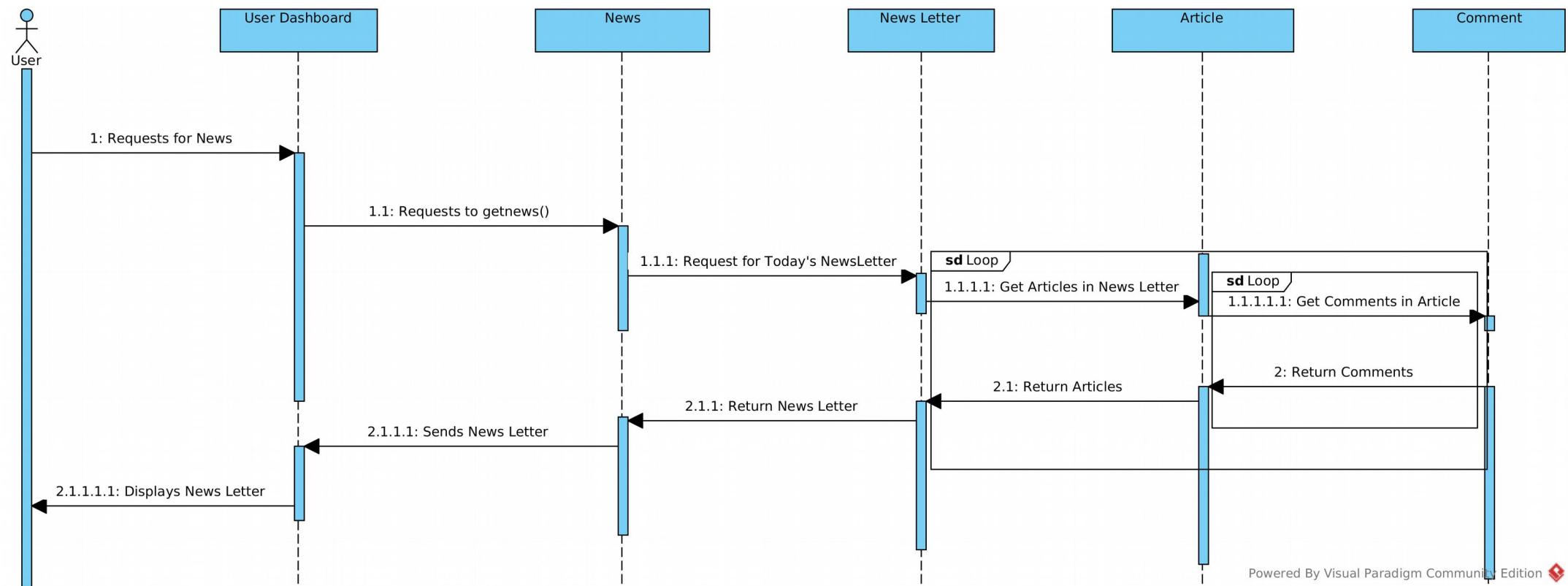
6. Sequence Diagram for Student request for grade report:



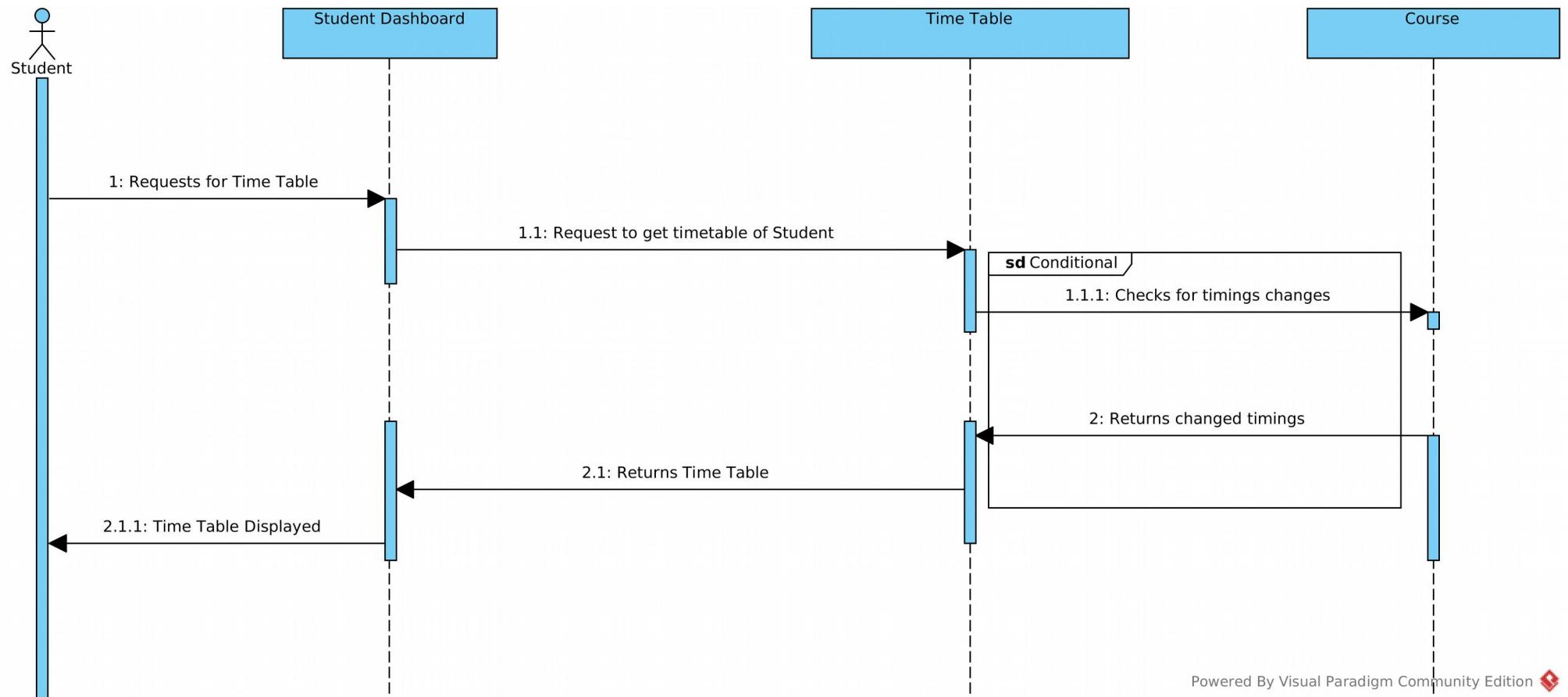
7. Sequence Diagram for Application of Leave form:



8. Sequence Diagram for Requesting News :



9. Sequence Diagram for Display TimeTable:



5. Design

Abstraction:

The current design is modelled in such a way that there is a separation between the interface and the implementation and only the essential information is provided to the outside world hiding the background details using the public private protected variables.

Separation of concerns:

The classes are designed such that any complex problem is broken down into smaller sub divisions hence making them easy to handle.

Modularity:

The classes are packaged into various packages and hence the data and function are compartmentalized, making it easier for maintenance.

Hiding:

The interfaces are build such that they are controlled and only display relevant information to the different types of users. For example in case of admin's manage user interface only the required information of update user/edit user is shown.

Extensibility:

Specialisation and aggregation of classes is used extensively keeping the future modifications in mind such that adding a new type of class/entity is not complex. For example if in case we want to add a timetable/ schedule for a different kind of users other than students/ faculty we already have a template in the form of the timetable class thereby making it easy for extending our system.

High cohesion:

Cohesion refers to the degree to which the elements of a module belong together, the system is designed is such a that the strength of relationship between pieces of functionality within a given module is high. For example the Finance module consists of all the classes which deals with the monetary of the institute ,thereby making the system robust and reliable and the elements of the module strongly related to each other.

Low coupling:

Coupling occurs when there are interdependencies between one module and another. The system is broken down into various packages in such a way that they interact with each other only through various interactions thereby ensuring low interdependencies between various modules and classes and changes in one place will not require changes somewhere else. For examples, if we change the branch of a student then we need to change only the student class and everywhere else we are using foreign key studentID to get the courses the student is enrolled to .

Law of Demeter:

The principle of least knowledge is followed since each class is only communicating either with a interface present between different classes or with its immediate friends and hence each unit only has limited knowledge about other units which is closely related to the current unit only.

Design Patterns Followed

The balance in the context of expected product evolution , the design patterns:

- Singleton Pattern: The creational pattern ,singleton pattern is used to control the creation of instance to just one . In our for example in case of the “Institute Money” class the singleton pattern is used to make sure that an instance of the class is created only once.
- Observer Pattern: The Behavioural pattern, observer pattern is used to address the problem associated with the assignment of responsibility objects and the manner in which communication is affected between objects, by enabling loose coupling between the publishers and subscribers. For example,
 - 1) From the class diagram we can see that the dashboard class is the subject and observes the interfaces like “news”, “messages” and the “users” interface and hence the dashboard class is loosely coupled to the observables like the news class, usermessages class and the user class and only know that they implement obverver interface
 - 2) From the class diagram we can see that Student dashboard is the subject and observes the courses offered class and the Fee card class observer courses interface and Feecard interface.

6. Reflection on various aspects of the design

Strong aspects:

- The Modularity of the Design where the system is broken down into various modules, thereby making it easy for maintenance.
- High cohesion is maintained between members of each module which gives the benefits like readability, maintainability , and reusability.
- Low coupling is maintained between individual modules such that testability and maintainability of the system is high.

Weak aspects:

- In classes like Batch, user List , mailingList , arrays of user id are stored as attributes, Since array is of fixed size, if we allocate more memory than requirement then the memory space will be wasted. And if we allocate less memory than requirement, then it will create problem. Moreover , the database will be complicated and it will become difficult to normalize and maintain.

