



 slington college
(इरिलिङ्टन कलेज)

Module Code & Module Title
CS5004NI Software Engineering

Assessment Weightage &
Type 35% Individual
Coursework

Year and Semester

2021-22 Autumn

Student Name: Pratiush Prasain

London Met ID: 20048959

College ID: NP01CP4S210210

Assignment Due Date: May 09, 2022

Assignment Submission Date: May 09, 2022

Title (Where Required):

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Acknowledgement

I would like to express my gratitude to Islington College for providing us with this fantastic opportunity to learn more about the many processes and phases of software development and application development. Throughout the course of completing my coursework, I ran into a number of issues. Without the help of my module leader and module tutor, Mr. Sanjeev Lama, I would have struggled to finish this report. He guided me through the issue and oversaw the completion of this report. I'd also like to express my gratitude to the module leader for his assistance. I'd also like to thank online technology for allowing me to investigate and learn more about this topic throughout my report.

Abstract

In this coursework, I'm supposed to construct a system for an application or software for a T-14 training academy which is specialized in football training.

Due to an inflow of new students and customer, T-14 training academy has been experiencing a number of issues recently because of the pandemic of covid. Because of the pandemic, T-14 training academy want to make their system online which include of registration of students, trade of the football kit as well as take a mock exam for the intermediate students.

The coursework has been completely constructed based on the scenario's requirements, with slight deviations when necessary. All relevant diagrams like Gantt Chart, Use case, Communication diagram and class diagram have been included in their entirety, and all sections of the report have been appropriately prepared. In summary, this paper details all of the strategies and processes involved in developing a good system for T-14 academy, from beginning to completion.

Table of Contents

1. Introduction.....	1
1. Amis and Objective	2
2. Methodology Used.....	3
Gantt Chart	5
3. Use Case Model	7
3.1. USE CASE DIAGRAM.....	9
4. High Level Use Case Description.....	10
5. Expanded Use Case	12
6. Communication Diagram	16
6.1. Drawing Collaboration Diagram:	17
6.2. Drawing Sequence Diagram:.....	19
7. Class Diagram.....	21
7.1. Completed Class Diagram.....	26
8. Development Process	27
9. Prototype.....	29
10. Conclusion.....	35
References.....	36

Table of Figure

Figure 1 Gantt Chart.....	6
Figure 2: Use Case Diagram.....	9
Figure 3: Collaboration Diagram.....	18
Figure 4: Sequence Diagram	20
Figure 5: Class Diagram	22
Figure 6: Adding Association between classes	23
Figure 7: Adding Inheritance between classes.....	24
Figure 8: Complete Class Diagram.....	26

1. Introduction

This coursework has given us a scenario that request to create online system for T-14 training academy. T-14 Training Academy is a training academy that specialize in football. The academy carries out various training programs focusing on the multiple age groups. For every age group, there are two types of training categories available: Basic or Intermediate. There are no specific criteria for joining Basic training. But to enroll in the intermediate training, there are some prerequisites that need to be completed. To join intermediate training, one needs to appear on some football IQ related examinations and must score certain points to be considered for other tests. T-14 Football Academy is looking for an online solution to manage their business, which includes the membership process and online exams.

According to the scenario, we, as a software engineer, are asked to design an online system for T-14 football Academy's business management, including features such as client registration, online tests, reporting, and more. The RUP methodology will be used to create this online system.

1. Aims and Objective

The aims and objective of this coursework is to:

- Gains a better understanding of the software development process when the design phase is completed.
- To gains the better understanding of the Gantt Chart and the methodology use to create it.
- To improve your capacity to design various types of diagrams on your own throughout future software development.

2. Methodology Used

The Rational Unified Process (RUP) is an object-oriented modeling software development process. The Rational Unified Process (RUP) methodology is created using an object-oriented approach, with UML (Unified Modeling Language) notation used to represent the processes in motion. It employs approaches and processes that have been proven in the marketplace. It is a labor-intensive approach that is best suited to large development teams and huge projects, but because it is very adaptable, it may be applied to projects of any size (testbytes, 2019).

The RUP (Rational Unified Process) paradigm provides a disciplined solution for project management, such as defining the tasks and responsibilities inside a software development organization. Some phases of RUP methodology are:

Initiation/Design: Its emphasis on the scope of the system. The system's basic requirements are translated into use cases at this level. The goal is not to close all of them, but simply those that are required to change public opinion. The phase is usually brief and is used to determine whether the project can be continued as well as the risks and costs of the previous one. For the client's approval, a prototype can be created.

Preparation: It focuses on the architecture. The preparation will be for the system design, in addition to the survey and/or documenting of use cases, in front of the system architecture, to review the project's business model, and to begin the version of the user manual.

Construction: It focuses on development. The physical creation of the software, production codes, and alpha tests begin during the construction phase. At the start of the changeover phase, beta testing were conducted.

Transition: It focuses on the application. The delivery ("deployment") of software occurs in this phase, which includes the execution of the deployment and delivery plan, as well as the monitoring and quality assurance of the software. Products (releases, versions) will be given, and customer satisfaction will be measured. During this stage, the users are also trained.

Features of RUP methodology:

- Full methodology with an emphasis on efficient procedures.
- Because components are reused, development time is reduced.
- Integration takes less time since it happens continuously throughout the software development life cycle.
- Improves risk management.

Gantt Chart

A Gantt chart is a form of bar chart that depicts a project schedule and highlights the dependencies between activities as well as the present status of the plan (Kashyap, 2022). Gantt charts are visual representations of tasks plotted against time. They represent crucial information in a project, such as who is assigned to what, task durations, and overlapping activities.

A Gantt chart adapts as the project progresses, providing an up-to-date project schedule and keeping everyone (teams, clients, and stakeholders) aware of the progress. Gantt charts also serve to replace meetings and improve other status reports. They make it simple for teams to understand work progress and discuss any issues that arise. Gantt charts are useful in project management because they allow you to visualize your project in relation to the time limit in which it must be completed.

3. Use Case Model

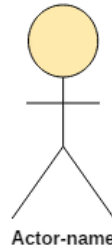
A use case is a description of a user's interactions with a system. A use case can define success and failure scenarios, as well as any essential variants or exceptions. With the help of a use case model tool, a use case can be written or visualized. Use Case Diagram captures the system's functionality and requirements by using actors and use cases. High-level functionalities and how a user would interact with the system are represented by use cases. The key ideas of Unified Modeling language modeling are use-cases (Walker, 2022). A Use Case is made up of use cases, people, or other objects that invoke the features known as actors, and the parts that are in charge of putting the use cases into action. Use case diagrams depict the live system's dynamic behavior. It represents how an external entity interacts with the system in order for it to function. External things that interact with the section of the system are visualized using use case diagrams.

Notations use in Use Case Diagram

Use-case: High-level functionalities and how the user will interact with the system are represented by use cases. A use case describes a specific feature of a system, component, package, or class. It's represented by an oval shape with the name of a use case written inside.



Actor: The actor is a component of the system that interacts with it. The best example of an actor is a user. An actor is a person or thing that starts a use case from outside its scope. It can be any aspect that can cause the use case to interact with it. In the system, one actor can be linked to several use cases.



Draw Use Case Diagram

To create a UML use case diagram, one must first thoroughly examine the entire system. We must learn how to use every feature that the system offers. After all of a system's functionalities have been identified, these functionalities are translated into numerous use cases for use in the use case diagram.

3.1. USE CASE DIAGRAM

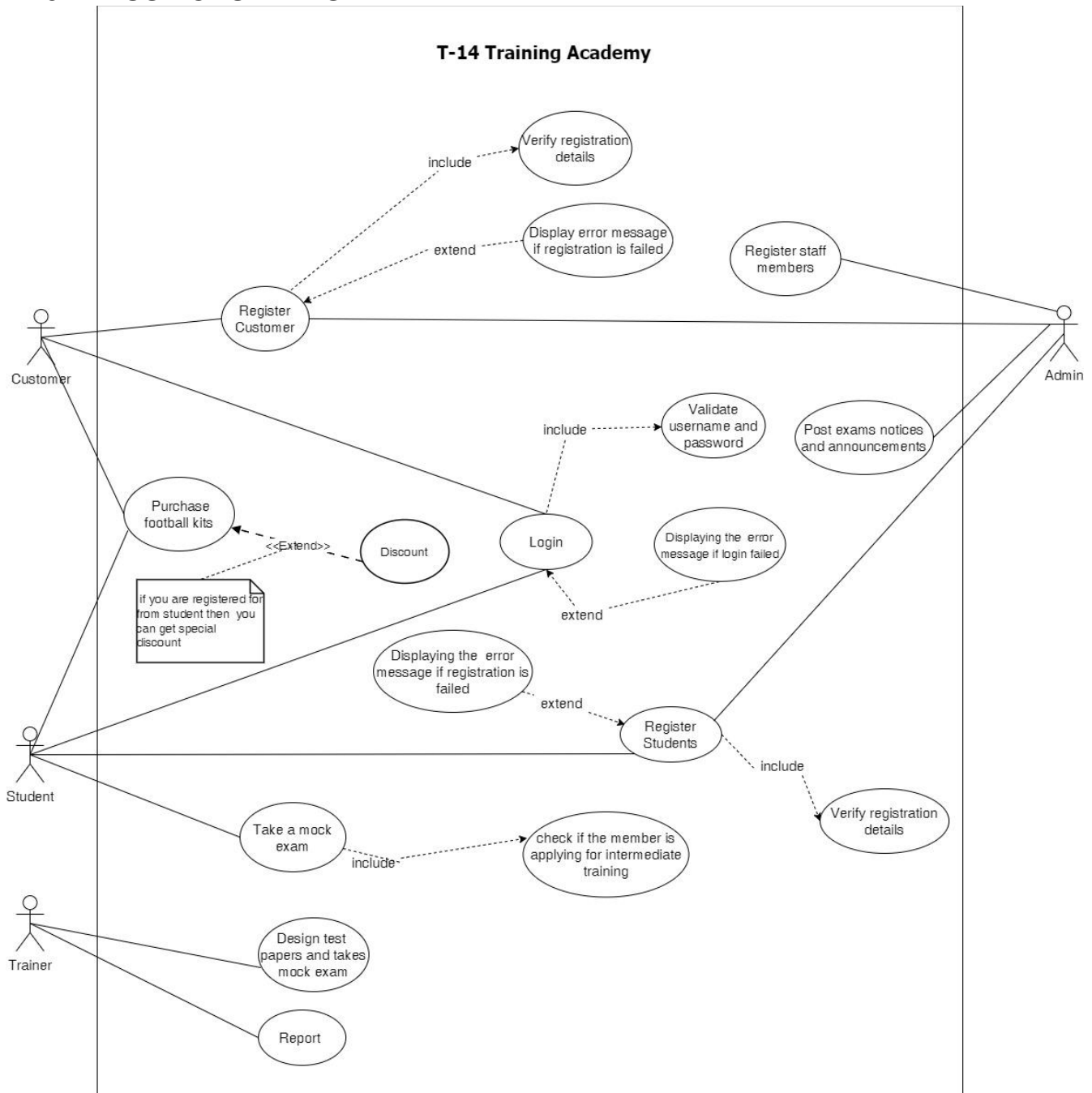


Figure 2: Use Case Diagram

4. High Level Use Case Description

The high-level use case is essentially a brief explanation of the task stated in unstructured prose of one or two paragraphs. It is used for just enough information to get a sense of the problem's complexity and to aid you in grouping related use cases for development in the Elaboration phase.

The high-level description that are written for the above use cases are shown below:

Use Cases	View Report
Actor	Students
Description	A student can view their progress of exam result.

Use Cases	Mock Exam
Actor	Students
Description	A student needs to pass an mock exam.

Use Case	Register Membership
Actor	Students, Customer
Description	Students/Customers need to register their personal information in T-14 academy.

Use Case	Payment
Actor	Student, Customer
Description	Student/ Customer choose the payment method for the payment of the football kit that they purchase from online store.

Use Case	Log in
Actor	Students
Description	Before entering in the system, student need to login first. Log in can be happened after filling up the registration form.

Use Case	Set Exam paper
Actor	Trainer
Description	Trainer set the exam papers for the students.

Use Case	Enroll the Staff
Actor	Admin
Description	Admin is the head of the system who can change and modify the announcement. Admin can enroll the staff.

Use Case	Store Customer Details
Actor	Trainer
Description	Trainer creates the database where the trainer stores the data of customers.

Use Case	Store Student Details
Actor	Trainer
Description	Trainer creates the database where the trainer stores the data of student.

5. Expanded Use Case

Student Register

Use Case: Student Register

Actor: Student

Description: A register form is filled by the students to make themselves registered in T-14 Training Academy. If the new students tries to join the academy then, he/she must fill up the registration form.

Course of Event

Action of Actor	System Response
Click the Register button.	
	Display the registration form
Fill up the online form with personal details.	
Submit the registration form	
	Verify the registration details.
	Store the details

Alternatives Cases:

Line 3: Use case ends, if user leaves any field blank.

Line 6: If user information is not valid then use case ends.

Use Case: Log in

Actor: Student

Description: Student can successfully log in to the system after he/she completely get registered.

Course of Event

Action of Actor	System Response
Fill up the registration form	
	Verifies the Details
	Validate the details of the student
Set the username and password	
	Successfully get in the system.

Alternative Cases:

Line 1: If registration failed, Use case ends

Line 4: If student give wrong username and password, use case ends

View Report

Use Case: View Report

Actor: Student

Description: A student can view their exam report. Report is available only for the students that are enrolled and registered.

Course of Event

Action of Actor	System Response
Log in to the system	
	Verifies the login
	Shows the student details
Input the ID, Name	
	Shows the report of the student.

Alternative Cases:

Line 1: If log in failed, use case ends.

Line 4: If entered ID and Name are incorrect then, use case ends.

Download Sample paper

Use Case: Download sample paper

Actor: Student

Description: A student can view and download the sample and practice question from the system once they logged in the system.

Course of Event

Action of Actor	System Response
Log in to the system	
	Verifies the login
	Shows the student details
Input the ID, Name	
Click on sample paper and download it	
	Generate a pdf file of sample questions.

Alternative Cases:

Line 1: If log in failed, use case ends.

Line 4: If entered ID and Name are incorrect then, use case ends.

6. Communication Diagram

Following the use-case diagram, high-level use case explanation, and expanded use case, the collaboration diagram is drawn. A collaboration diagram is an interaction diagram that displays information in a similar way to a sequence diagram but focuses on object relationships. Objects are displayed on communication diagrams with association connections connecting them. Messages are added to associations and shown as short arrows pointing in the message flow's direction. A numbering scheme is used to show the order of the communications (sparxsystems, 2022).

Component of Communication Diagram

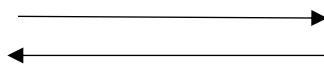
Rectangle — This symbol symbolizes the application's objects. To summarize, these are the components that communicate in order to carry out orders or keep the program or system running.



Lines — The flow of communications throughout the system is directed by this communication diagram feature. It also displays the relationships between objects within the system.



Arrows — This component depicts the message flow direction.



Number: The order in which the messages are sent is represented by the numbers in this type of diagram.

1.1,1.2,1.3

6.1. Drawing Collaboration Diagram:

In the Unified Modeling Language, this is an illustration of the relationships and interactions between software objects (UML). These diagrams can be used to depict the dynamic behavior of a certain use case and define each object's purpose.

Notations use in collaboration diagram

Objects: The object's name and class are used to represent itself.

Actor: The actor is the most important element in the collaboration diagram since it initiates the interaction. Each actor is assigned a part and given a name. The use case is started by a single actor.

Link: The link is an example of association in which the items and actors are linked together. It depicts a connection between the items that send messages. A solid line is used to depict it.

Message: It is a form of communication between things that conveys information and includes a sequence number in order for the activity to occur. A labeled arrow next to a link is used to signify it (javatpoint, 2022).

How to draw collaboration diagram

- Study the behavior for which implementation and realization are necessary.
- Discover the structural pieces of collaborative functionality, such as class roles, objects, and subsystems.
- Choose from system, subsystem, use case, and operation as the context for an interaction.
- Consider the several scenarios that could arise.

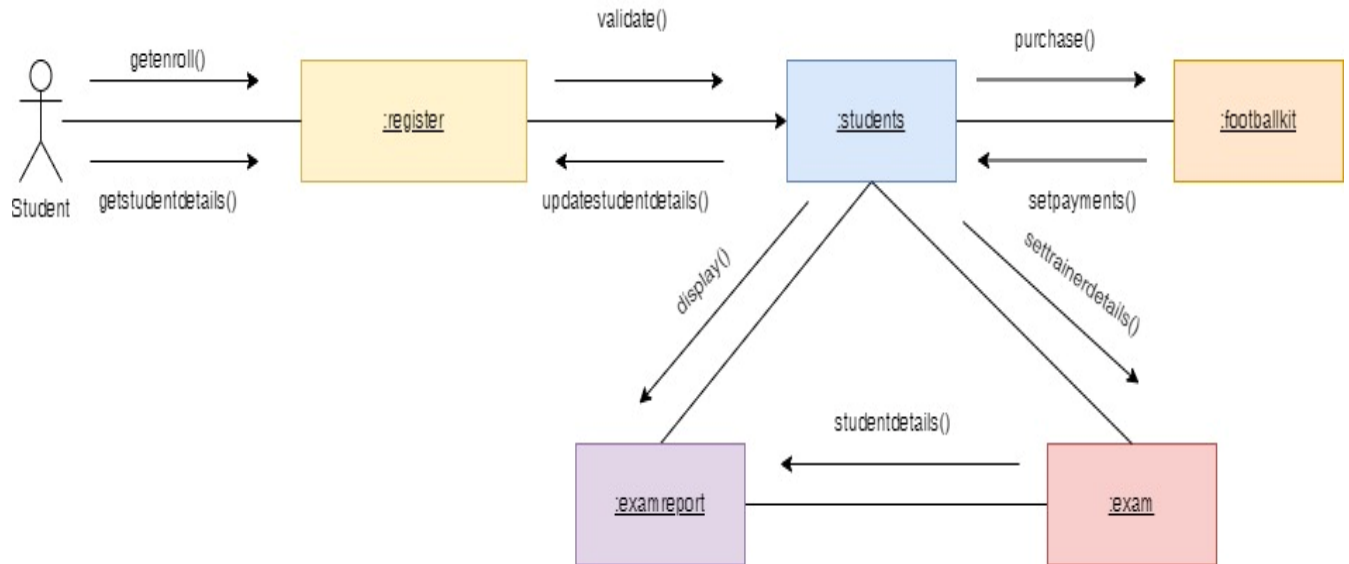


Figure 3: Collaboration Diagram

6.2. Drawing Sequence Diagram:

Notations use in sequence diagram

Synchronous message symbol: A solid line with a solid arrowhead is used to represent this. When a sender must wait for a response to a message before proceeding, this symbol is used. Both the call and the response should be depicted in the diagram.



Asynchronous message symbol: A solid line with a lined arrowhead is used to represent this. The sender of asynchronous communications does not need to wait for a response before proceeding. In the diagram, only the call should be included.



Asynchronous return message symbol: A dashed line with a lined arrowhead is used to represent asynchronous return message (lucidchart, 2022).



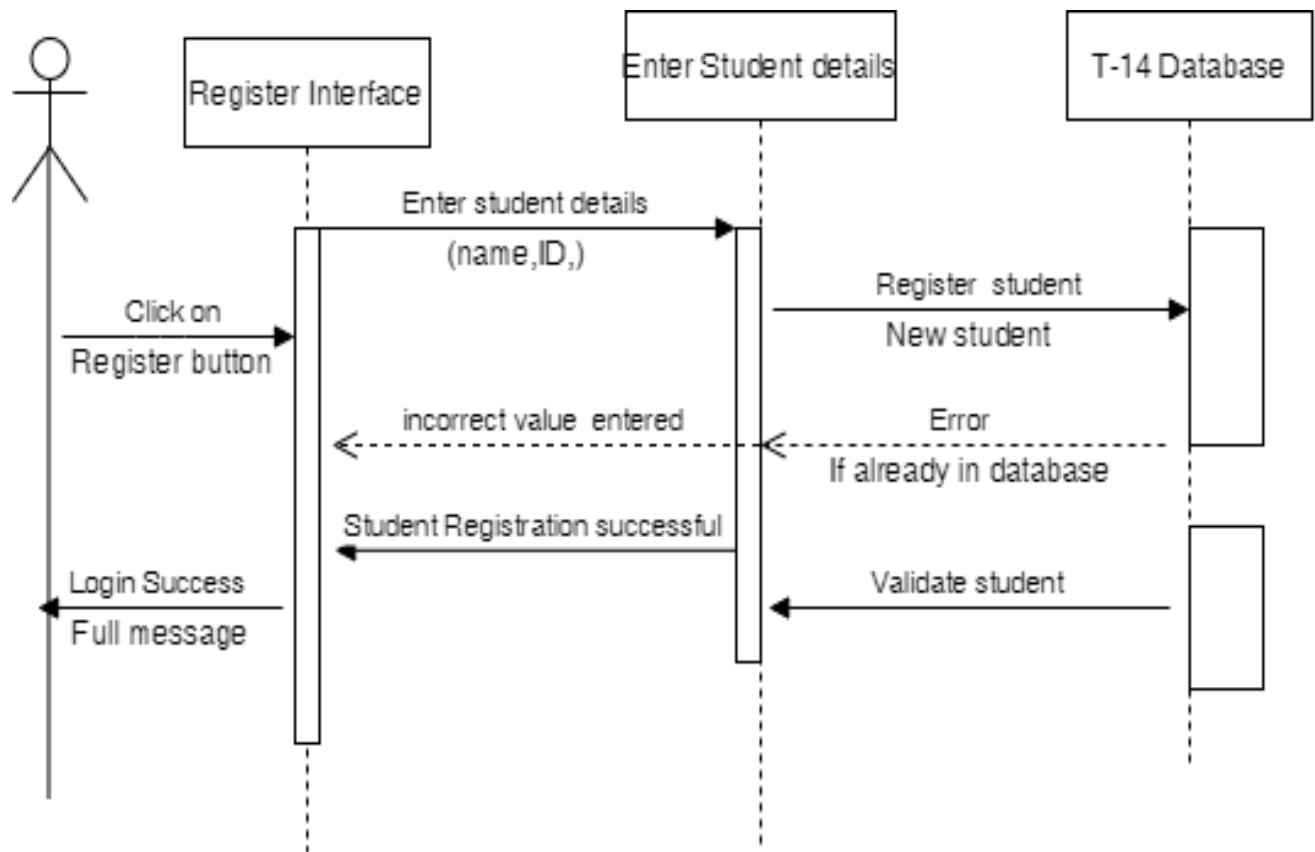


Figure 4: Sequence Diagram

7. Class Diagram

The class diagram is one of the forms of UML diagrams that is used to depict the static diagram by mapping the system structure using classes, attributes, relations, and operations amongst the various objects. The first partition of a class diagram contains a Class name, which is the name of the class or entity that is participating in the activity, the second partition contains class attributes, which show the various properties of the class, the third partition contains class operations, which shows the various operations performed by the class, and the fourth partition contains relationships, which shows the relationship between two classes (Pedamkar, 2022).

The existence of a relationship between the classes is required in a class diagram. Some of the relationship in class diagram are explained below:

Association: An association class is a class that exists between two other classes in an association connection. The association relationship could be linked to the association class to get further information about the relationship.

Aggregation: By putting several elements together in this form of interaction, a more complex object is created. Aggregation defines the interaction between distinct groups of items. The objects' integrity is preserved, and the control object determines the combined objects' reaction.

Composition: The whole-part connection is represented by this type of aggregation. The component classifier lifetime is influenced by the overall classifier lifetime in this case. The composition connection in a class represents a strong life cycle. In most cases, data flows in only one direction. A solid line is usually used to indicate it.

Unique classes of the system are:

- Login
- Students
- Registration Membership
- Customer
- Football Kit
- Staff
- Payment
- Admin
- Report
- Exam

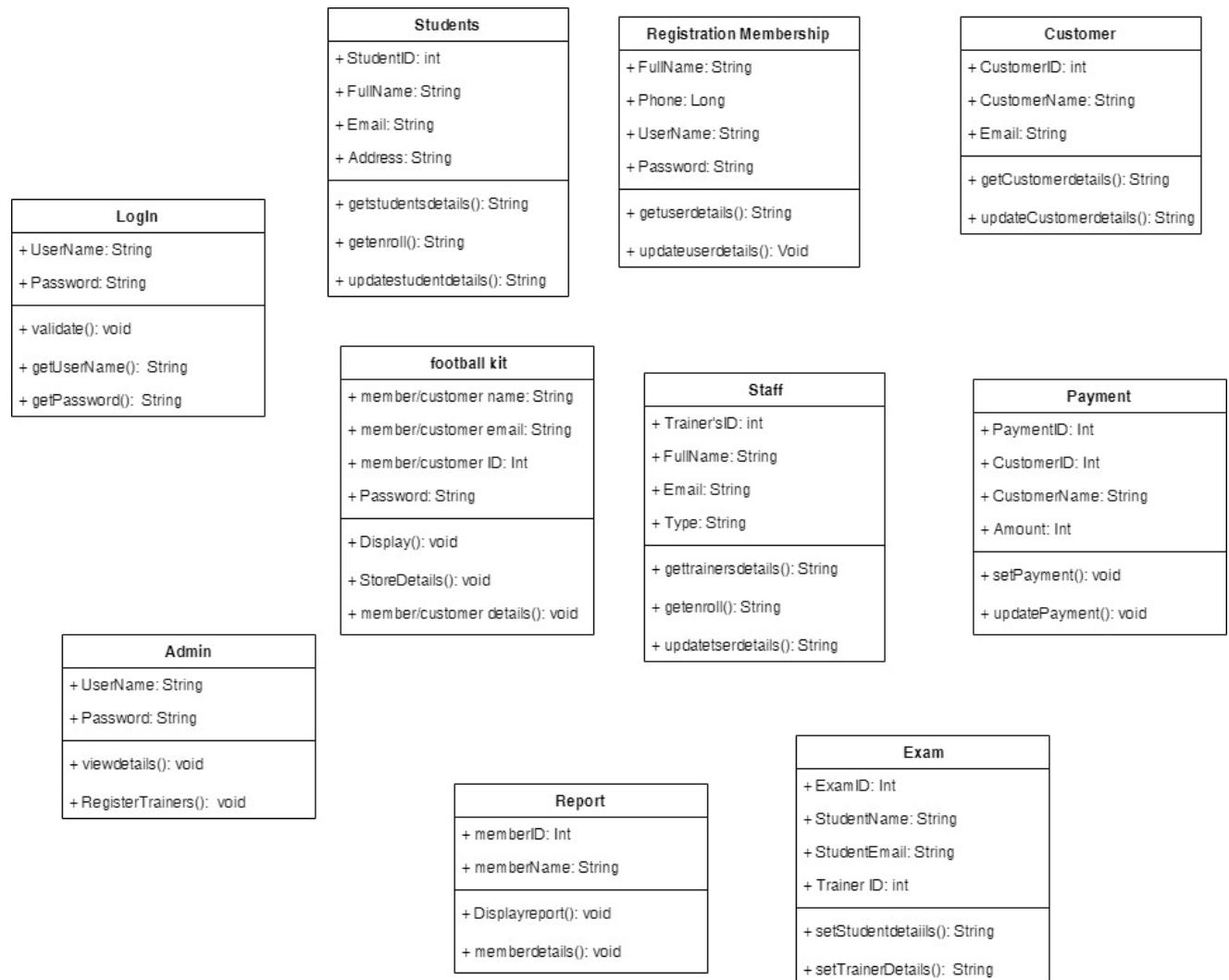


Figure 5: Class Diagram

Adding Association between classes

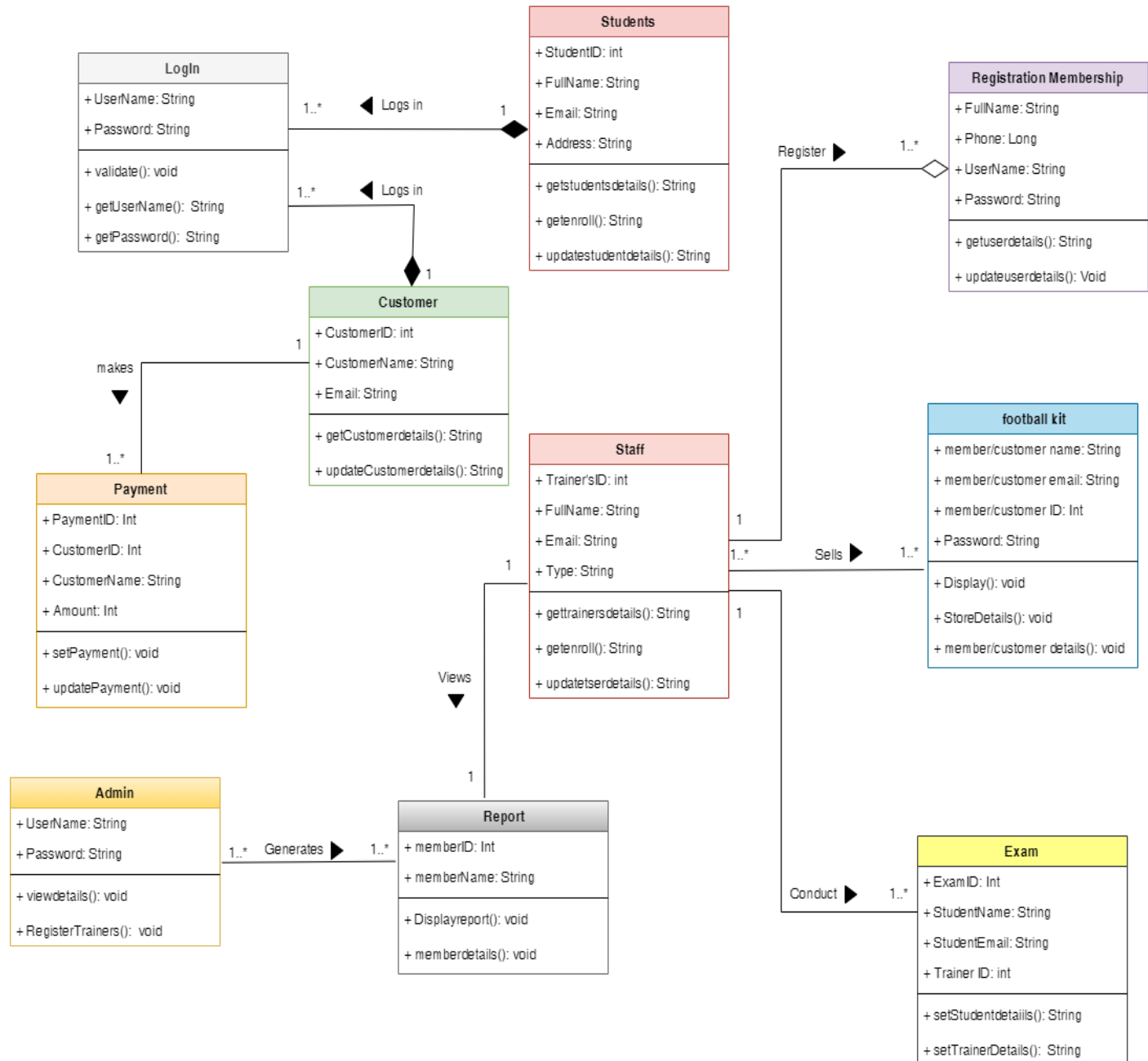


Figure 6: Adding Association between classes

Adding Inheritance among the classes

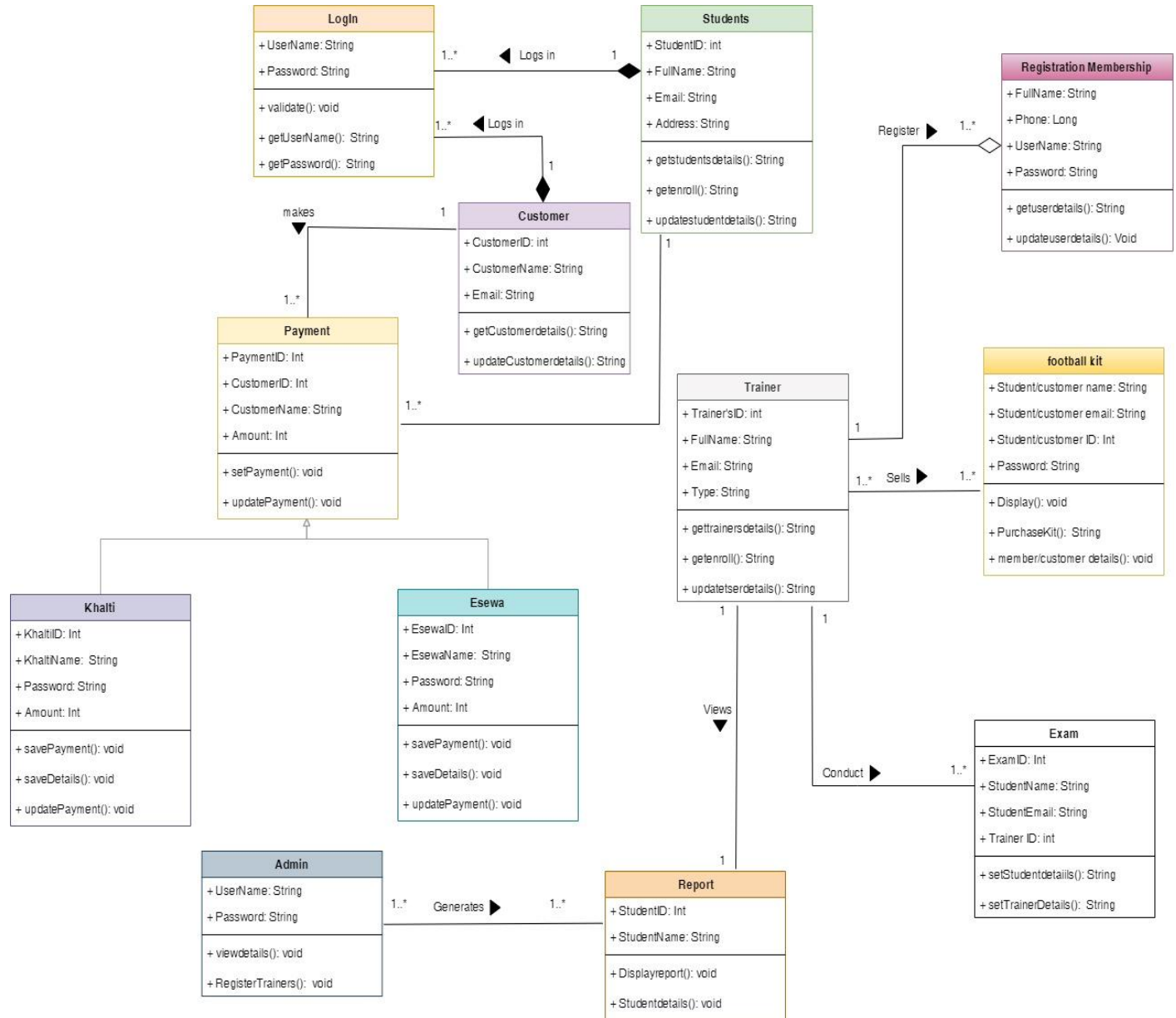


Figure 7: Adding Inheritance between classes

Table of Use Case and its respected Domain Classes

Use Case	Domain Classes
Login	Student, Admin
Registration Membership	Customer, Student
Register Form	Customer, Student
View Report	Student
Generate Report	Admin, Trainer
Payment	Customer, Student
Purchase Football kit	Customer, Student
Conduct Exam	Trainer, Student
Enroll Staff	Admin

7.1. Completed Class Diagram

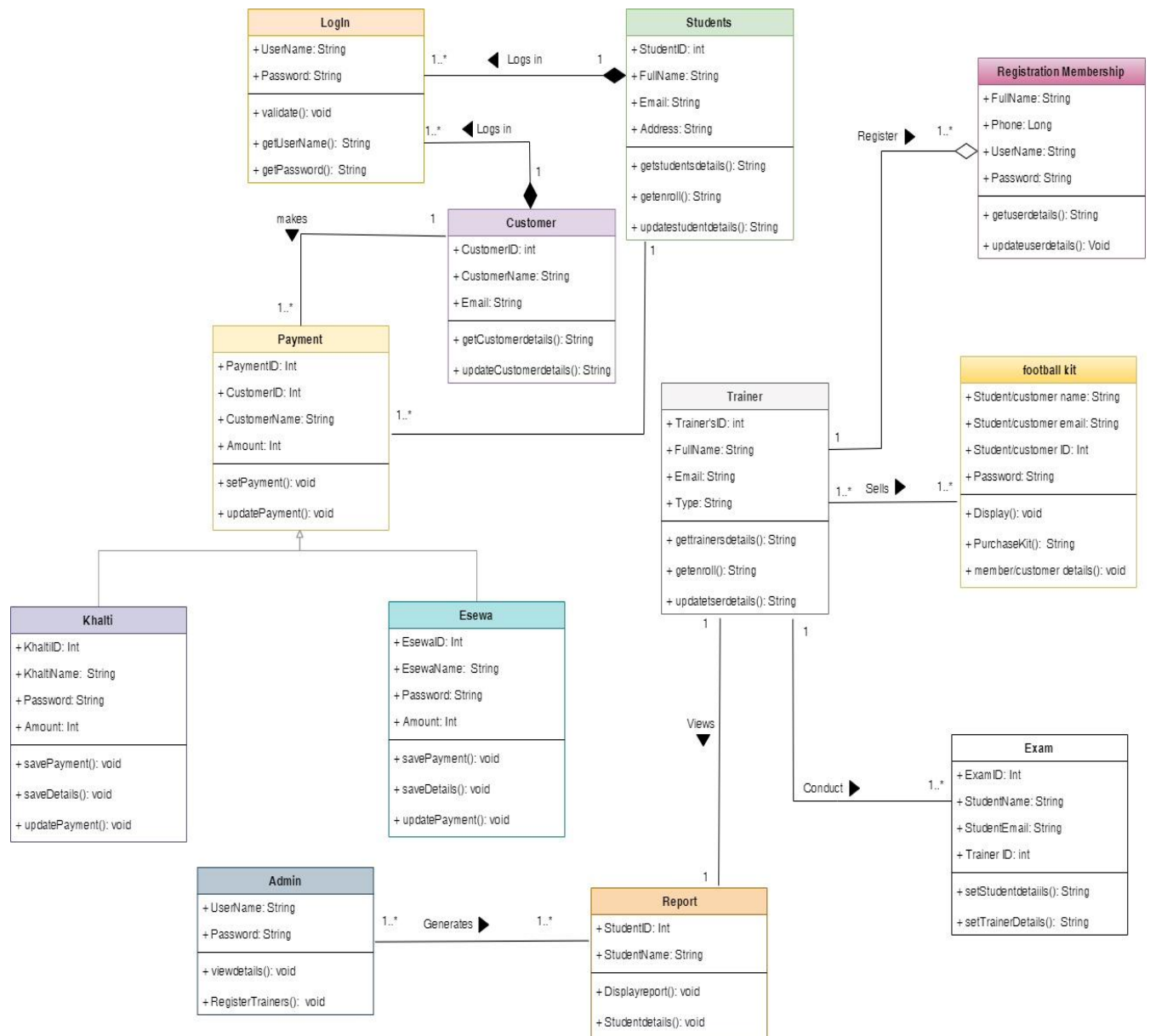


Figure 8: Complete Class Diagram

8. Development Process

All the above architecture and software design work was completed during the elaboration phase. The software design method included communication diagrams, sequence diagrams, and use case diagrams. The designing process is not yet complete because I am using RUP approach to prepare this application. Because RUP is an iterative software design approach, all of these designing and figuring processes may be repeated in subsequent rounds. As a software engineer, I'd like to demonstrate how I'd like to go in the development process.

Some major actions are:

Coding: The first iteration of coding begins when the initial phase of elaboration is accomplished. In the future development phase, the code will be developed. Coding takes the most time out of all the construction steps. In the future, I intend to write code using node js, a fantastic programming platform. The code must be written in an error-free and straightforward manner. To improve the program's efficiency, superfluous code should be removed.

Testing: After the success of the coding phase, I plan to conduct all the testing like unit testing, system testing and integration testing one by one. Testing is very useful for checking the overall performance and checking the system's durability, maintenance as well as sustainability of the t-14 software. All the testing like black box, white box, grey box testing will be done in the required stage of the development process.

Maintenance and support: A good software always need to be keep maintained and sustain. Without regular maintenance, the application could decrease substantially. I want to provide more upgrades and maintenance features in the future to add new features and fix errors. Because we are using the RUP methodology, the maintenance stage may begin concurrently with the other stages.

User Manual: After the development of the system, a well prepared and well managed user manual is created for the new trainer who enroll in the system. User manual is created for the trainer who are unaware about the use of the system.

User Review: The T-14 academy system is not only for the trainers. It is also for the student and customers. It is very necessary to know the user review. User review also helps to enhance the system in future.

9. Prototype

i. Log in

A Web Page

https://www.T-14academy.co.np/log-in/page

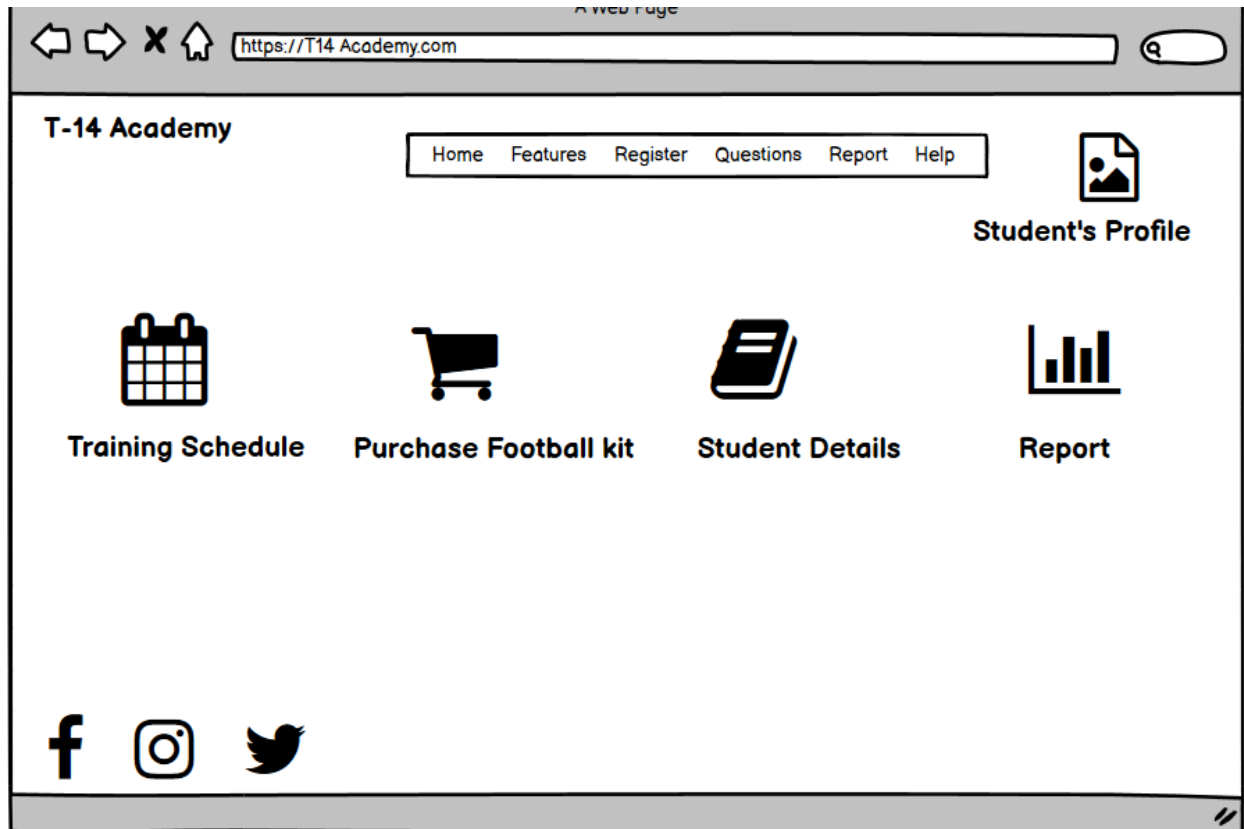
T 14 Training Academy

User Name:

Password:

[Forget Password](#)

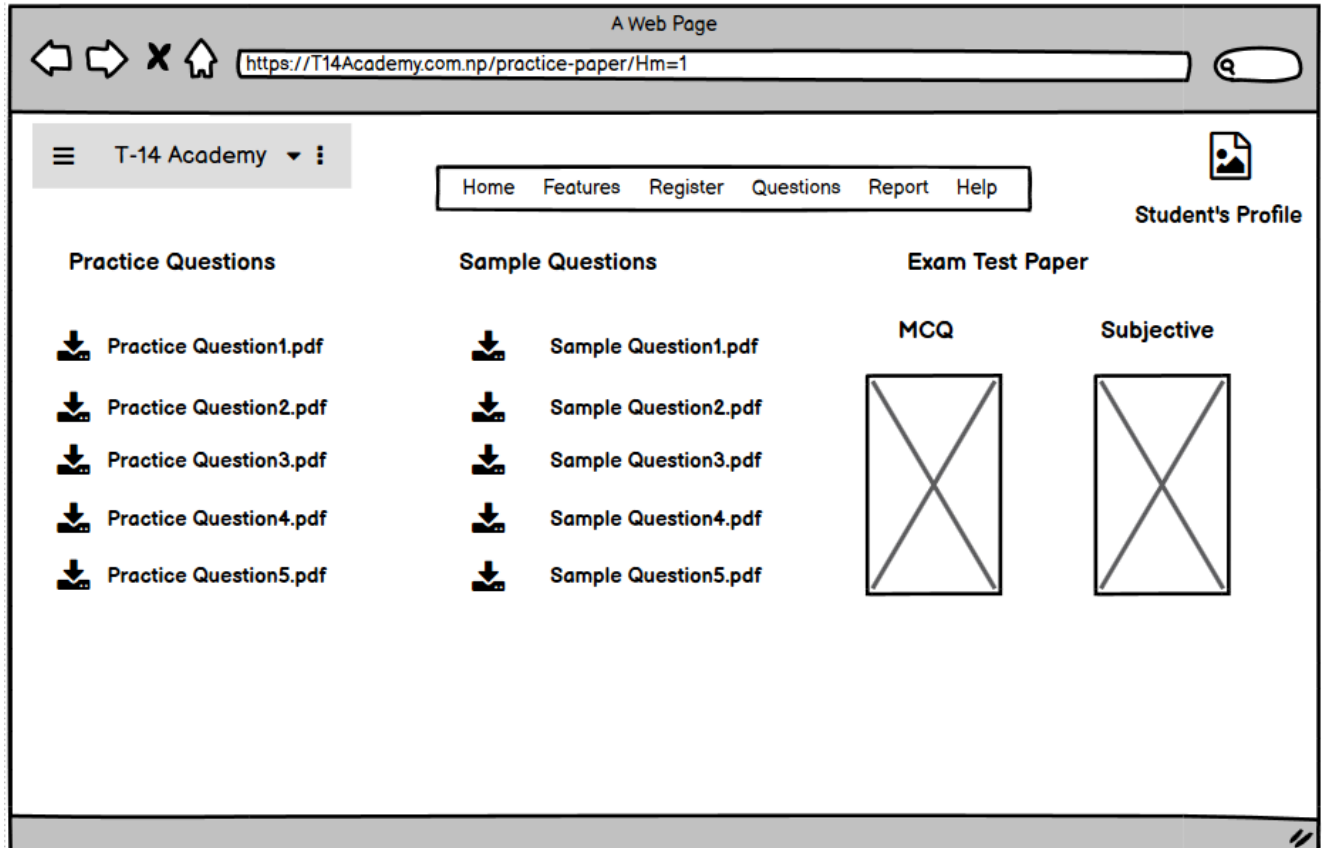
ii. Home page



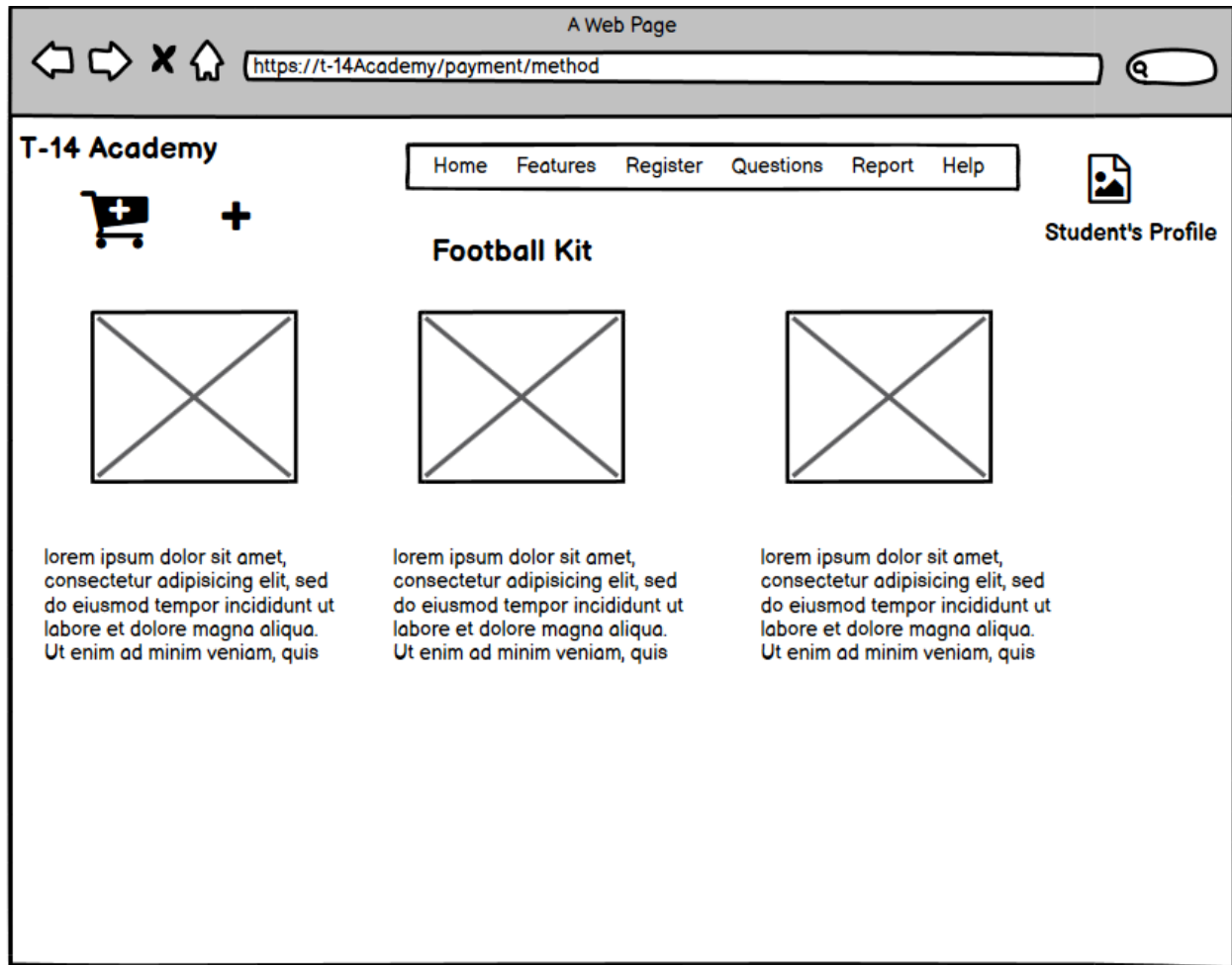
iii. Registration form

The screenshot shows a web browser window with the title 'A Web Page'. The address bar contains the URL 'https://T14academy.com.np/register-Form,M=1'. The page content is titled 'T 14 Registration Form' and features a navigation menu with links: Home, Features, Register, Questions, Report, and Help. The registration form includes the following fields and elements:

- First Name:** A text input field with the placeholder 'Enter your First Name'.
- Last Name:** A text input field with the placeholder 'Enter your Last Name'.
- Country:** A dropdown menu with the placeholder 'Country'.
- Date of Birth:** A text input field with the placeholder 'DD/MM/YY' and a calendar icon.
- Phone:** A text input field with the placeholder 'Enter your Phone Number'.
- Gender:** Two radio buttons labeled 'Male' and 'Female'.
- Address:** A text input field with the placeholder 'Enter your Address'.
- User Name:** A text input field with the placeholder 'Enter your User Name'.
- Password:** A text input field with the placeholder 'Enter your Password'.
- Terms and Privacy condition:** A checkbox labeled 'Terms and Privacy condition'.
- Register:** A button labeled 'Register'.
- double-click to edit:** A button labeled 'double-click to edit'.

iv. Sample and Practice questions

v. Purchase Football Kits



vi. Payment Method

A Web Page


https://www.t14academy.com.np/payment methods


T-14 Academy

Home Features Register Questions Report Help

Student's Profile

Payment Options

 **khalti**
DIGITAL WALLET

 **eSewa**
www.esewa.com.np

UserID

User Password

10. Conclusion

The coursework was emphasizes on object orientation development . The purpose of this coursework is to develop an online system for T-14 Academy. Because everything was new to me, it was tough for me to finish all of the figures and diagrams. Creating a Gantt chart, sequence diagram, use case diagram, and class diagram was a difficult task.

The module leaders and teacher taught me about the RUP technique, which will come in handy when planning future projects. I also learnt how to utilize a Gantt chart to visualize the progress of a project over time. Knowing how to construct use cases, communication diagrams, and class diagrams will surely assist me in developing additional products in the future.

References

- javatpoint, 2022. *UML Collaboration Diagram*. [Online]
Available at: <https://www.javatpoint.com/uml-collaboration-diagram>
- Kashyap, S., 2022. *What is Gantt Chart?*. [Online]
Available at: <https://www.proofhub.com/articles/gantt-charts>
- lucidchart, 2022. *UML Sequence Diagram Tutorial*. [Online]
Available at: <https://www.lucidchart.com/pages/uml-sequence-diagram>
- Pedamkar, P., 2022. *Class Diagram*. [Online]
Available at: <https://www.educba.com/class-diagram/>
- sparxsystems, 2022. *UML 2 Tutorial - Communication Diagram*. [Online]
Available at: <https://sparxsystems.com/resources/tutorials/uml2/communication-diagram.html>
- testbytes, 2019. *What is Rational Unified Process Methodology?*. [Online]
Available at: <https://www.testbytes.net/blog/rational-unified-process/>
- Walker, A., 2022. *UML Use Case Diagram: Tutorial with EXAMPLE*. [Online]
Available at: <https://www.guru99.com/use-case-diagrams-example.html>