Module Code & Title

CS5002NI SOFTWARE ENGINEERING

Assessment Weightage & Type

35% Individual Coursework

Year and Semester

2021-22 Spring

Student Name: Punam Thapa Magar

London Met ID: 20048968

College ID: np01cp4s210273

Assignment Due Date: 9th May 2022

Assignment Submission Date:9th May 2022

Title (Where Required): Methodologies

Word Count (Where Required): 4145

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

Table of Contents

List of Tables

List of Figures

1. Introduction

This is the second coursework for the module “Software Engineering” in which we are assigned a task to design an online system for T-14 Training Academy to manage their business operation even in the current situation of the Covid-19 pandemic.

To operate the business of T-14 Training Academy through an online medium, a system named **T-14 Training Academy** is developed. The utilization of the system can be done to register memberships, design exam test papers and practice test papers, enrol staff members, purchase football kits, generate reports, post-exams notices and announcements and take a mock exam. Comprehensively the functionality of these functions will be given in the framework. To carry out the work for fostering the framework, Unified Software Development Process (USDP)/Rational Unified Process (RUP) ought to be utilized with the unified Modelling Language (UML). The Gant chart is made to show the time of consummation of the entire errands following the RUP methodology. Then, the use case is developed to gather every one of the prerequisites for the development. The entire coursework is focused and object-oriented. The collaboration diagram, sequence diagram, and class diagrams are developed to meet the requirements of the coursework. Towards the finish of the coursework, a prototype of the software has been developed to outline the standpoint of the software of the T-14 Training Academy. To develop the diagrams the use of draw.io has been done.

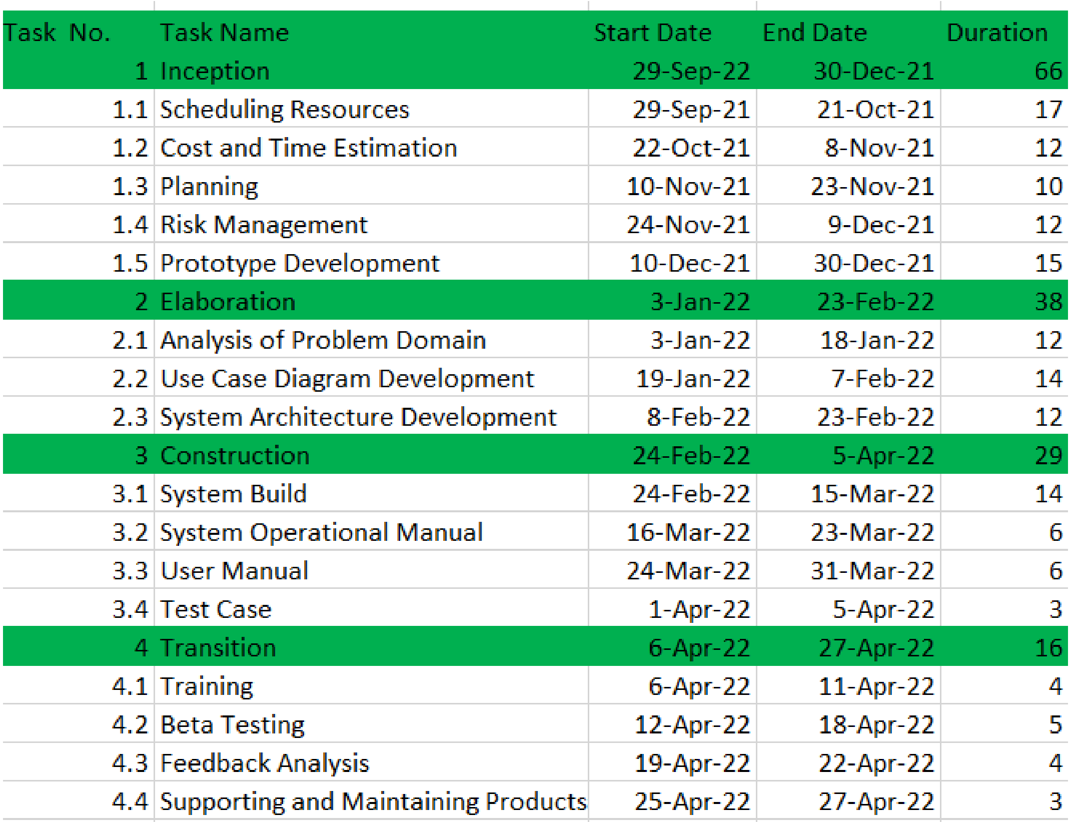
The main aim of this coursework is to create a framework that can meet all the requirements of the business. The objective of this coursework is to comprehend and acquire information on Structured Software Engineering to meet every one of the prerequisites of the task.

2. Gantt chart

A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes, although they are particularly useful for simplifying complex projects. Project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, as well as dependencies, scheduling, and deadlines, including how much of the task is completed per stage and who is the task owner. This is useful to keep tasks on track when there is a large team and multiple stakeholders when the scope changes (apm, 2022).

The Gantt Chart below was created utilizing the RUP methodology using the four phases i.e., Inception, Elaboration, Construction, and Transition which includes activities like Project Planning, Risk Management, System Design, System Testing, Beta Testing, etc. The chart was made using the online Gantt chart-making tool, 'TeamGantt'.

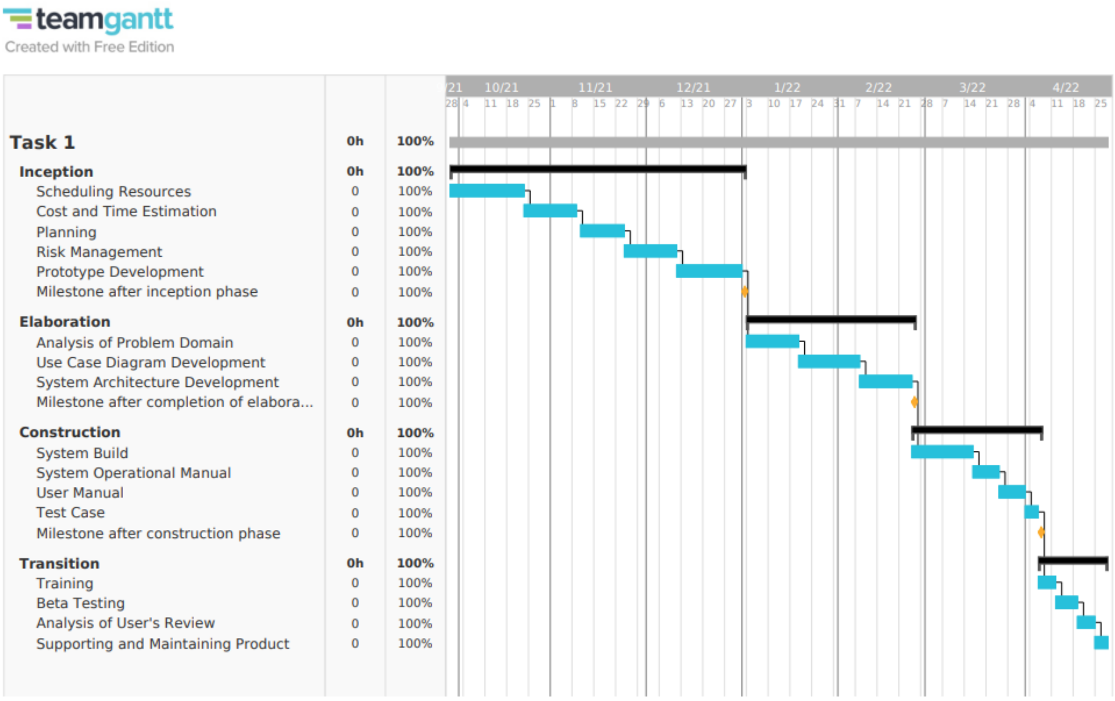
Gantt chart reflecting methodology



Task No.	Task Name	Start Date	End Date	Duration
1	Inception	29-Sep-22	30-Dec-21	66
1.1	Scheduling Resources	29-Sep-21	21-Oct-21	17
1.2	Cost and Time Estimation	22-Oct-21	8-Nov-21	12
1.3	Planning	10-Nov-21	23-Nov-21	10
1.4	Risk Management	24-Nov-21	9-Dec-21	12
1.5	Prototype Development	10-Dec-21	30-Dec-21	15
2	Elaboration	3-Jan-22	23-Feb-22	38
2.1	Analysis of Problem Domain	3-Jan-22	18-Jan-22	12
2.2	Use Case Diagram Development	19-Jan-22	7-Feb-22	14
2.3	System Architecture Development	8-Feb-22	23-Feb-22	12
3	Construction	24-Feb-22	5-Apr-22	29
3.1	System Build	24-Feb-22	15-Mar-22	14
3.2	System Operational Manual	16-Mar-22	23-Mar-22	6
3.3	User Manual	24-Mar-22	31-Mar-22	6
3.4	Test Case	1-Apr-22	5-Apr-22	3
4	Transition	6-Apr-22	27-Apr-22	16
4.1	Training	6-Apr-22	11-Apr-22	4
4.2	Beta Testing	12-Apr-22	18-Apr-22	5
4.3	Feedback Analysis	19-Apr-22	22-Apr-22	4
4.4	Supporting and Maintaining Products	25-Apr-22	27-Apr-22	3

Figure 1:Gantt chart reflecting methodology

Gantt chart using methodology



Task 1	**0h**	**100%**
Inception	**0h**	**100%**
Scheduling Resources	0	100%
Cost and Time Estimation	0	100%
Planning	0	100%
Risk Management	0	100%
Prototype Development	0	100%
Milestone after inception phase	0	100%
Elaboration	**0h**	**100%**
Analysis of Problem Domain	0	100%
Use Case Diagram Development	0	100%
System Architecture Development	0	100%
Milestone after completion of elabora...	0	100%
Construction	**0h**	**100%**
System Build	0	100%
System Operational Manual	0	100%
User Manual	0	100%
Test Case	0	100%
Milestone after construction phase	0	100%
Transition	**0h**	**100%**
Training	0	100%
Beta Testing	0	100%
Analysis of User's Review	0	100%
Supporting and Maintaining Product	0	100%

Figure 2: Gantt chart using methodology

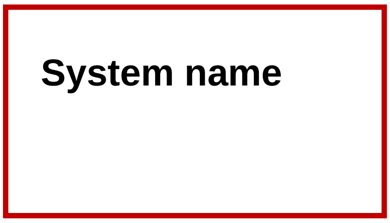
3. Use Case Model

The Use-case model is characterized as a model which is utilized to show how users associated with the system tackle an issue. Accordingly, the use case, model characterizes the user goal, the interactions between the system and the user, and the system’s performance required to meet these targets. Various model elements are contained in the use-case model, for instance, actors, use cases, and the association between them.

The major notations of the use case model are given below:

- **System**

With the help of the rectangle, we can draw the boundaries of the system, which includes use-cases. We need to put the actors outside the system's boundaries (javaTpoint, 2022)

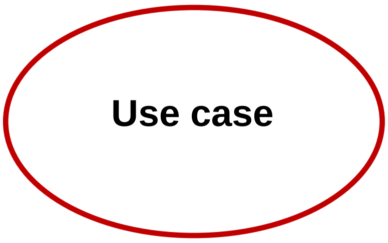


Use Case

Figure 3: System

- **Use-case**

With the assistance of the ovals, we can draw the use-cases. With the verb, we need to name the ovals to address the functions of the system.



Use case

Figure 4:Use-case

- **Actors**

Actors mean the system's users. On the off chance that one system is the actor of the other system, then with the actor stereotype, we have to tag the actor system. Besides, actors are represented by the human stick sort of picture, who are the clients of software systems.

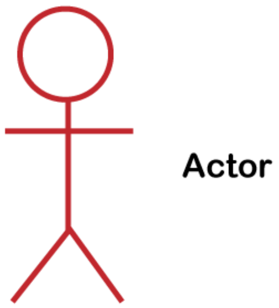
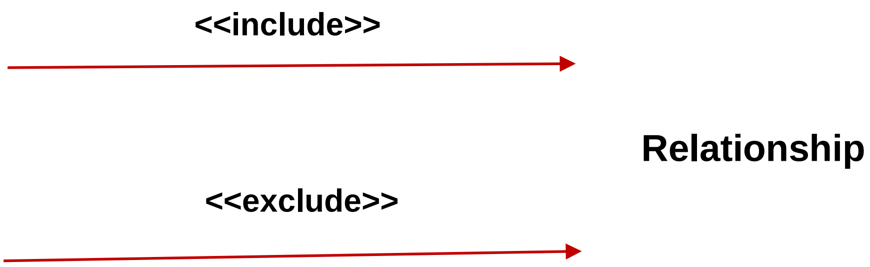


Figure 5:Actor

- **Relationship**

With the simple line, we can represent relationships between an actor and use cases. For relationships between use-case, we use arrows that are labeled either "extends" or "uses". The "extends" relationship shows the alternative options under the specific use case. The "uses" relationship shows that a single-use case is required to accomplish a job (javaTpoint, 2022).



3.1. Use case Diagram

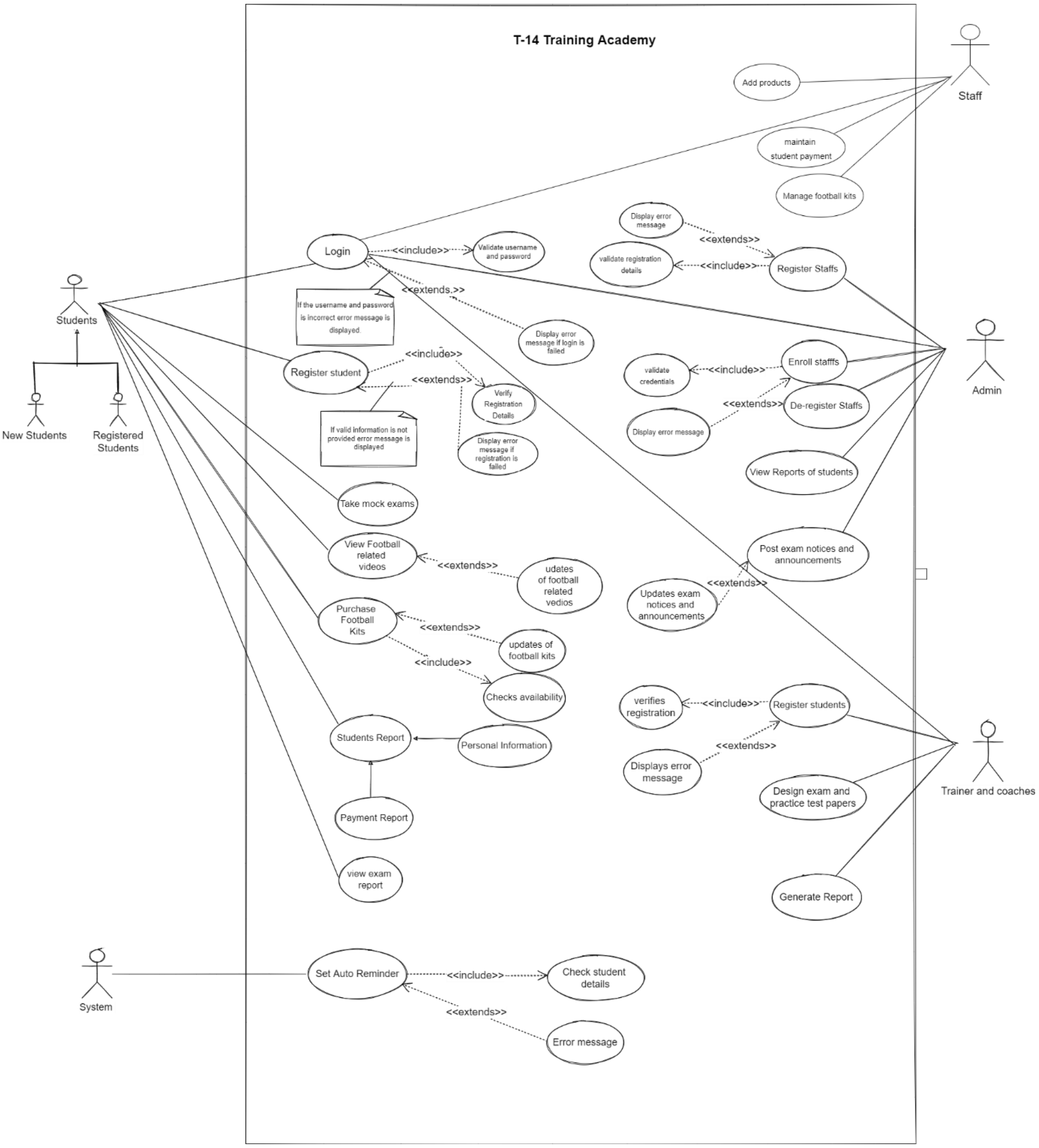


Figure 6: A use case diagram for T-14 Training Academy

The above use case diagram portrays all the software system’s use cases and actors. There are three actors i.e student, admin, and trainer/coaches, and staff. It additionally exhibits the include and excludes relationship. As seen in the diagram over, the system required several functions.

3.2. High-level use case description

High-level use case description is a short non-detailed description of the use case. High-level use case descriptions of all use cases of this project are given below:

3.2.1

Table 1. Use case 1

Use Case:	Login
Actor:	New students/Registered students, Admin and Trainer/coach
Description:	Registers students, admin, and trainers/coaches should log in to get access to the T-14 Training Academy.

3.2.2

Table 2. Use case 2

Use Case:	Register student
Actor:	New students
Description:	The new students who have not registered before filling up the registration form and log in to get access to the T-14 Training Academy.

3.2.3

Table 3.Use case 3

Use Case:	Take mock exams
Actor:	New students/ Registered Students
Description:	After login, both the new students and registered students are required to take a mock exam after becoming a student at a T-14 Training Academy.

3.2.4

Table 4. Use case 4

Use Case:	View Football related videos
Actor:	New students/Registered students
Description:	The students have access to view the football-related videos which were generated by the trainer.

3.2.5

Table 5. Use case 5

Use Case:	Purchase football kits
Actor:	New students/Registers students
Description:	Students are given updates on the football kits and check their availability so that students can borrow them.

3.2.6

Table 6.Use case 6

Use case:	Student Report
Actor:	A new student/registered student
Description:	The student’s report is generated so that he/she can check all his or her details related to payment, tasks, and personal information.

3.2.7

Table 7. Use case 7

Use case:	View reports
Actor:	A new student/registered student
Description:	The students can view their reports

3.2.8

Table 8.Use case 8

Use case:	Register staff
Actor:	Admin
Description:	Admin can register the staff.

3.2.9

Table 9. Use case 9

Use case:	Enroll staff
Actor:	Admin
Description:	Admin can enroll the staff.

3.2.10

Table 10.Use case 10

Use case:	De-register staff
Actor:	Admin
Description:	Admin can de-register the staff.

3.2.11

Table 11. Use case 11

Use case:	View reports of the student
Actor:	Admin
Description:	Admin views the overall report of all activities in the system performed by the student.

3.2.12

Table 12. Use case 12

Use case:	Post exam notices and announcements
Actor:	Admin
Description:	Admin post the updated exam notices and announcements to the students.

3.2.13

Table 13.Use case 13

Use case:	Register students
Actor:	Trainer and coach
Description:	Trainer verifies the student registration in the system once the registration form is submitted in the system by student. If the registration requirements are not met, the system will show an error message.

3.2.14

Table 14.Use case 16

Use case:	Design exam and practice test papers
Actor:	Trainer and coach
Description:	In the T-14 academy, trainers and coaches design all the exam and practice test papers.

3.2.15

Table 15. Use case 18

Use case:	Generate report
Actor:	Trainer and coach
Description:	Trainers and coaches generated the overall report of all activities in the system to track down all the records.

3.2.16

Table 16. Use case 16

Use case:	Add products
Actor:	Staffs
Description:	Staffs can add required products in the academy.

3.2.17

Table 17. Use case 17

Use case:	Maintain student payments
Actor:	Staff
Description:	The record of the payments made by the students are managed and stored b the staffs.

3.2.18

Table 18. Use case 18

Use case:	Manage football kits
Actor:	Staff
Description:	Staff manages all the football kits at the T-14 Training Academy.

3.2.19

Table 19. Use case 19

Use case:	Set auto reminder
Actor:	System
Description:	The system sets an auto-reminder to the students a week before their membership expires. It also notifies students half an hour before their reserved time. Also, if there is any kind of offer or discount in academy than students are to be informed about the offer or discount details.

3.3. Expanded Use case Description

3.3.1 Use case: Register student

Actor: New Student, admin

Purpose: Registering a student

Overview: First of all, a student fills up the registration form as an underlying step to be required in the T-14 Training Academy. After submitting the form with all the expected details, the system illuminates the confirmation about the activity. Once the admin investigates, he/she verifies the enlistment of a student in the system. Then, students will be informed about the registration endorsement and can log in to the T-14 Training Academy subsequently.

A typical course of Events:

Table 19. Expanded use case description of register

Actor Action	System Response
1. New student fills out the form to register and confirms the registration.	
	2. Notifies the admin about a new registration.
3. Admin sees the registration and student details and confirms the registration.	
	4. Notifies the confirmation of the student.
	5. Provides the username and its password to the student as a login detail.
6. The student gets a registration confirmation notification and login details. He/she can log in and get access to the system from now on.	

Alternative:

Line 5: If the admin does not confirm the registration, the student won't be registered into the system and therefore, the system won't provide login details to the customer. Use case terminated.

2.3.2 Use case: Generate report

Actor: Trainer and coach

Purpose: Generate reports of the students

Description: The trainer and coaches are allowed to generate report of the students after the examination given by the student in the system.

A typical course of events:

Table 20. Expanded use case of generate report

Actor Action	System Response
1. The student applies for registration by filling out the registration form.	
	2. System approves the registration of the student.
3. The student logins into the academy by typing his/her username and password.	
	4. Give access to the academy.
5. Student takes the mock exam.	
6. Trainers and coaches evaluate the marks obtained by the students in the mock test.	
7. Trainer and coaches sort through the marks and the student details and compile them.	
8. Trainer and coaches save the marks obtained by the student for future reference.	

	9. The marks and details of students are sent to the system.
	10. The system thoroughly generates the report.

Alternative actions:

Line 9: If the trainer and coaches do not send the report to the system. Use case terminated.

4. Communication Diagram

Communication diagrams, also known as collaboration diagrams, are similar to UML sequence diagrams showing how objects interact. A communication diagram is an extension of an object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, the communication diagram shows the messages the objects send each other. (visual-paradigm, 2022).

- **Register Student**

Since we needed to make a communication chart for every one of the extended use case descriptions. We utilized the extended use case definition of the use case **“Register Student”** guideline. Bit by Bit, the communication diagram was developed.

Step 1:

We should initially characterize all the domain classes in a given use case. The expanded level description of the characterized use case can be utilized to observe domain classes. For the “Register Student” use case, the accompanying domain classes have been identified.

1. StudentRegister
2. Student

Step 2:

The domain class objects are then drawn. Every one of the domains that have been characterized should be addressed by (:) and encased in a rectangle box.

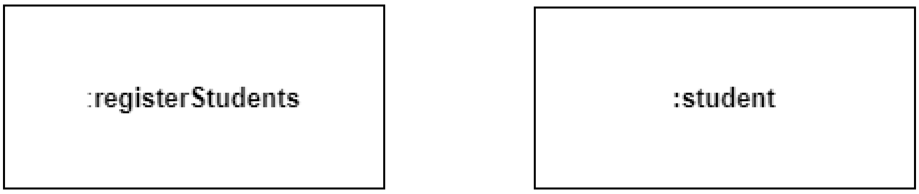


Figure 7: Communication Diagram- Domain objects

Step 3:

Following that, we should make a control object with the similar name as the use case.

:registercontroller/system

Figure 8: Communication Diagram-Control Object

Step 4:

Likewise, a boundary object should be formed. With the expansion of UI at the top, it will have the same name as the use case.

:RegisterstudentUI

Figure 9:Communication Diagram-Boundary object

Step 5:

It is mandatory to draw the actors involved.

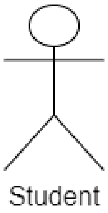


Figure 10: Communication Diagram-Actor(s)

Step 6:

The boundary object is connected to the actor, and connection between domain classes and control objects are laid out by using association relations.

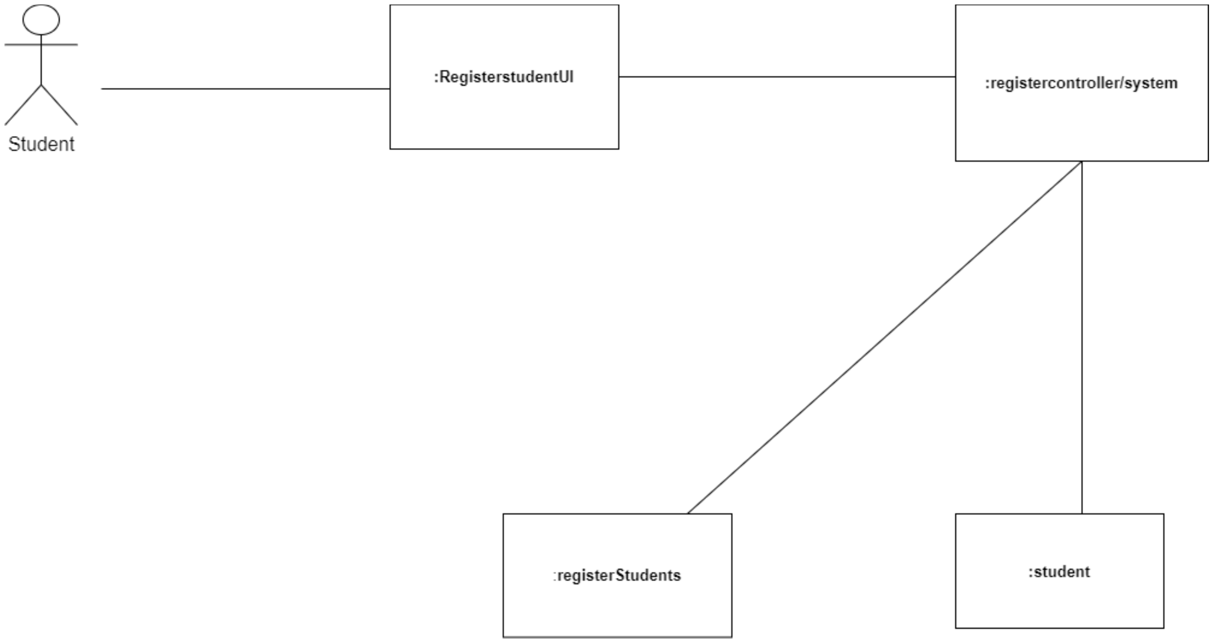


Figure 11:Communication Diagram- Objects

Step 7:

At long last, messages and required methods for passing data are added.

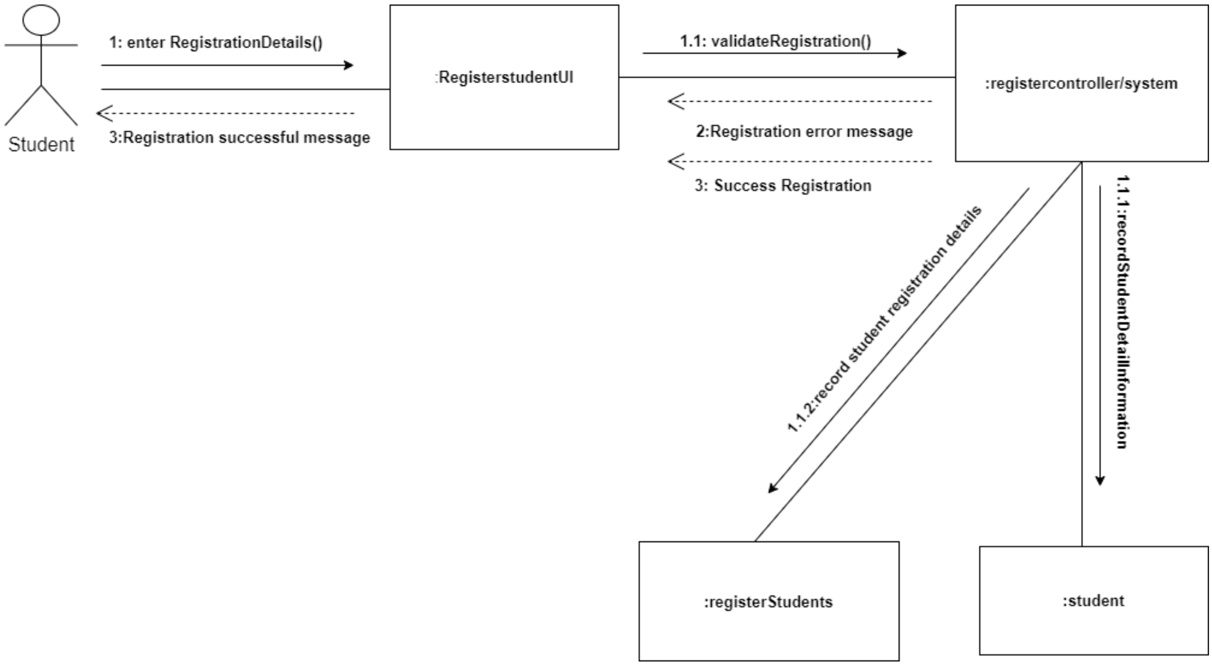

Student
1: enter RegistrationDetails()
3:Registration successful message
:RegisterstudentUI
1.1: validateRegistration()
2:Registration error message
3: Success Registration
:registercontroller/system
1.1.2:record student registration details
1.1.1:recordStudentDetailInformation
:registerStudents
:student

Figure 12: Communication Diagram

5. Sequence Diagram

Sequence diagrams, commonly used by the developers, model the interactions between objects in a single-use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed. In simpler words, a sequence diagram shows different parts of a system working in a 'sequence' to get something done. A sequence diagram is structured in such a way that it represents a timeline that begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows (creately, 2022).

The various parts of a sequence diagram are as follow:

1. Lifeline Notation
2. Actors
3. Activation Bars
4. Message Arrows

The purpose of sequence diagram is:

- Model high-level communication between active objects in a system.
- Model the Interaction between object instances inside a collaboration that understands a use case.
- Model the interaction between objects inside a collaboration that understand an activity.

There are some of the main benefits of using UML sequence diagrams. They are as per following:

1. Helps to find structural, connection point, and logic problems early.
2. It can be utilized to archive the unique view of the system design at different levels of abstraction, which is frequently challenging to extricate from the static outline or even the total source code.

The drawbacks of the sequence diagram are as mentioned below:

1. In the instance of many lifelines, the sequence diagram can get more intricate.
2. The erroneous outcome might be created if the request for the progression of messages changes.

A sequence diagram for use case T-14 Training Academy:

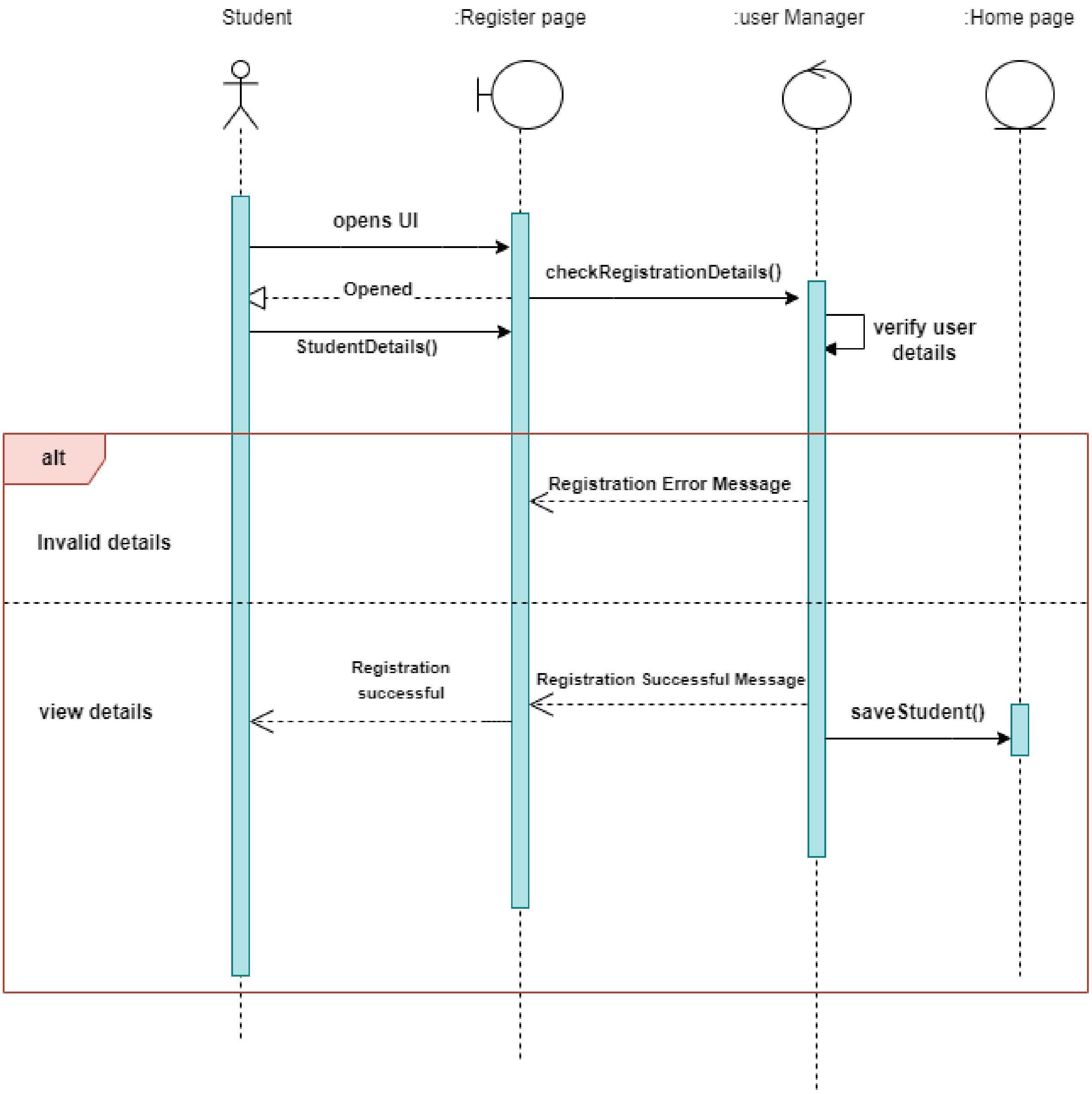


Figure 13. Sequence Diagram

6.Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Its purpose is to provide a basic notation for other structure diagrams prescribed by UML and is helpful for developers and other team members too (visual-paradigm, 2022). The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with an object-oriented language. It shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram (tutorialspoint, 2022)

The class diagram has class, its attributes, and methods. They are represented as below:

Class Name
Attributes
Methods

Below are the relationships which exist in a class diagram.

1. Association
2. Generalization
3. Aggregation
4. Composition

The steps to be encountered while making a class diagram are as follow:

- Right away, domain classes of the separate use cases need to be distinguished.
- Then, a class diagram for the domain classes is drawn with a class name, attributes and the methods.
- Connection between one class to another is recognized.

- Whenever required, cardinality ought to be portrayed.

Use case and their respective domain classes

Table 21: Class Diagram- Domain classes of the use cases

Use cases	**Domain classes**
Register	Student, Registration
Login	Student, login
Take mock exams	Student
View Football related videos	Student
Purchase Football kits	Student
Student Report	Student
View exam report	Student
Register student	Trainer
Enroll staffs	Admin
Post exam notices and announcement	Admin
Generate report	Trainer
Design exam and practice test papers	Trainer
Register staff	Admin
De-register staffs	Admin
View Reports of student	Admin

Class Diagram for T-14 Training Academy

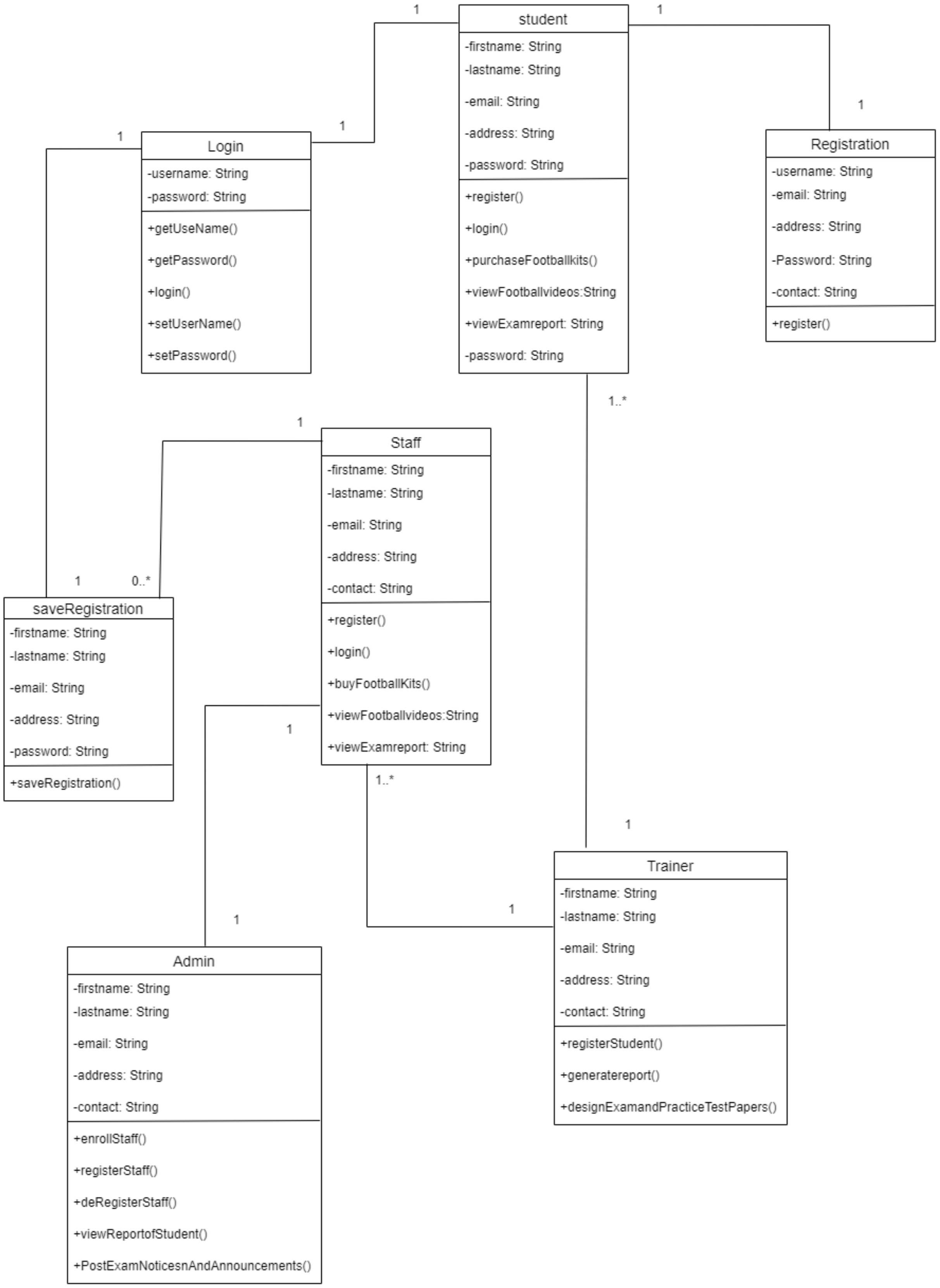


Figure 14: Class Diagram

7. Further Development

The coursework expected us to assemble an internet-based framework for T-14 Training Academy. We are utilizing the RUP methodology to create a project for "T-14 Training Academy". The project is still in progress, and we have just gotten to the extent that the RUP methodology's elaboration process. Up to this point, we've constructed the use case, communication, sequence, and class diagrams for the project. By initiating these procedures, we can now start to build the general system design.

This project required a ton of project analysis, research, requirement analysis, crude thoughts, planning, development, and testing of the system. The system was made however it tends to be improved in the further advances that will be taken sooner rather than later as many updates will be presented. It tends to be improved by expanding the nature of coding and, design, adding a lot more features, and better testing to fix the bugs and blunders.

The program was made utilizing the Unified Modelling Language (UML) and the Rational Unified Process (RUP). From the inception phase to the elaboration phase to the building phase to the transfer phase, the development process was made stride by step. Our software engineers and developers planned data models and programming codes for the application utilizing the concluded class outline created during the elaboration process. Our development team utilized the introduced and settled use case diagrams, use cases, communication, series, and class diagrams to develop working models for the system build.

For this project, Flutter was picked as the base application SDK, it conveys unmatched elite execution applications because of its cross-execution abilities. Our designers can utilize Flutter to assemble a codebase for a particular reason, which they can then tweak to address the issues of our clients. The application could be effectively altered to run on various platforms, including IOS, Android, and desktop, utilizing Flutter. Since it is adaptable and embraces an open-source ecosystem, the DART programming language is utilized for the programming language itself. It is the only language that can be utilized to make Flutter applications.

Furthermore, we pick IOS as our base stage for the customer adoption of the application since it is considered simpler for virtualizations, testing, troubleshooting, and prototype creation. As the integrated production environment, IOS Studios was utilized. It's a solid IDE that paces up the creation of IOS applications for cell phones. At last, due to its basic UI and solid turn of events, and debugging capabilities, Visual Studios Code was picked as the essential code proofreader. The staff and Admin page were fabricated utilizing a mix of HTML, CSS, XML, and JavaScript. At long last, Visual Studios Code was picked as the primary code editor because of its straightforward UI and solid creation and debugging capabilities. Moreover, the application is worked considering high attachments and how coupling for legitimate and all round-shaped code creation. Also, this empowers productive modularity of functions and methods, which can then be dispersed among qualified developers in view of their programming skills.

Our independent team of testers will direct various examinations utilizing these experiments as a part of the project's testing process. In the wake of recognizing device errors, unit testing, smoke testing, integration testing, and regression testing are directed independently as per testing methodologies' basic principles. Furthermore, white box testing, black-box testing, and grey box testing methodologies are used to recognize extra variations. Following the fulfillment of the application's blunder check testing, a prototype model of the software is constructed and introduced to our clients and their discussion groups to record their assessments and further evaluate the program's details conformance, execution, and wanted quality.

On April 06, the project's last stage, Transition, will start. With our clients' endorsement, different preparation protocols and regimens are created to teach employees on the most proficient method to manage, operate, and support the application. Similarly, beta testing of the application will start on April 12, 2022, to guarantee and record input on the pre-released application's functionality and activity in a real-world setting. The feedback is handled and tried to assist the program with advancing much further. Our clients get the application software, alongside the necessary documentation and upkeep, after it has been effectively dealt with and finished.

Besides, different phases of planning are being carried out simultaneously since the project is being planned to use the RUP/USDP methodology. Numerous tasks can cover because of the coordinated and dynamic plan of this methodology. As recently referenced, our group of negotiators and programmers will organize a gathering with our clients in the wake of accepting their feedback and surveys.

Prior to being re-released to the consumer, the created prototype is tested once more utilizing a few proposed test cases, and their criticism is assessed. The client manual and working manual will be refreshed, and their suggestions will be applied. These phases of inception, elaboration, design, and transformation will repeat before a content program is constructed that meets the concurred models of our advancing teams and clients. From that point forward, our client gets the authority and completed project, which will go on until the agreement between our clients, T-14 Training Academy, and my undertaking group terminates.

8. Prototype

Prototype is the diagrammatical depiction of the actual programming system and a fundamental type of the software that will be made, tending to the system starting phases and elements. The software will appear to be the prototype that has been expected for the system. A couple wireframe methods of the "T-14 Training Academy" are made to arrange project models.

1. Homepage

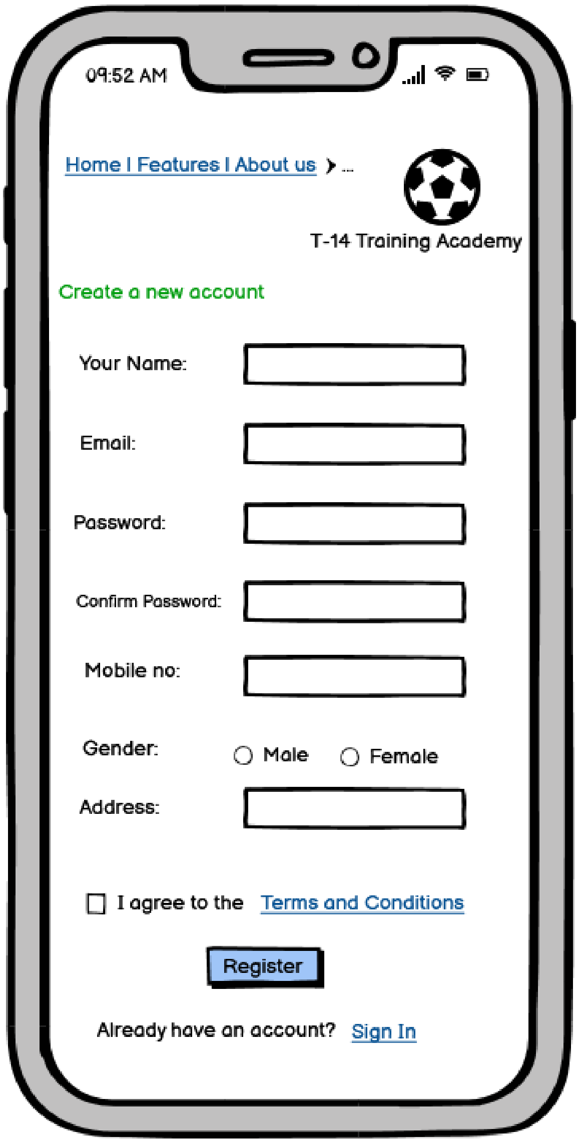
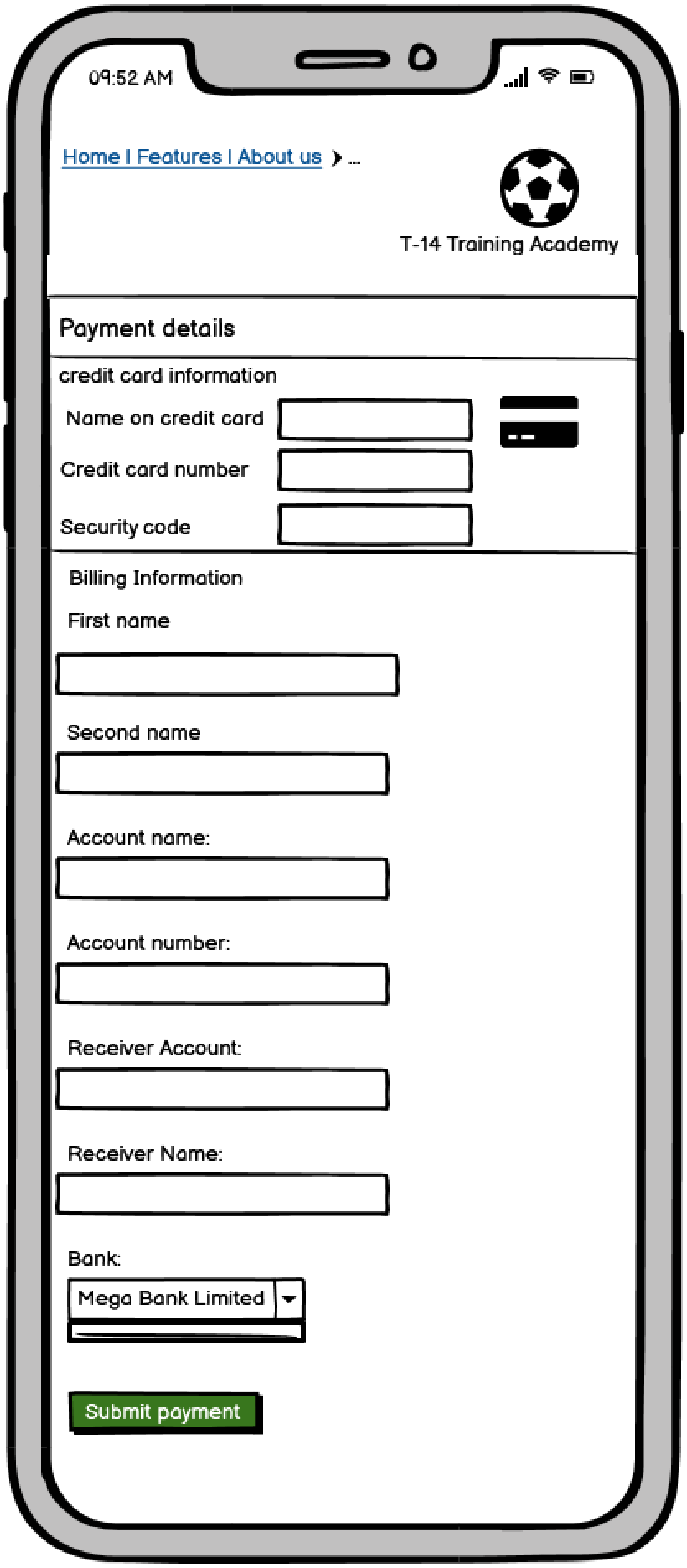


Figure 15: Homepage

2. Payment



Bank: Mega Bank Limited

Figure 16: Payment

3. Landing page

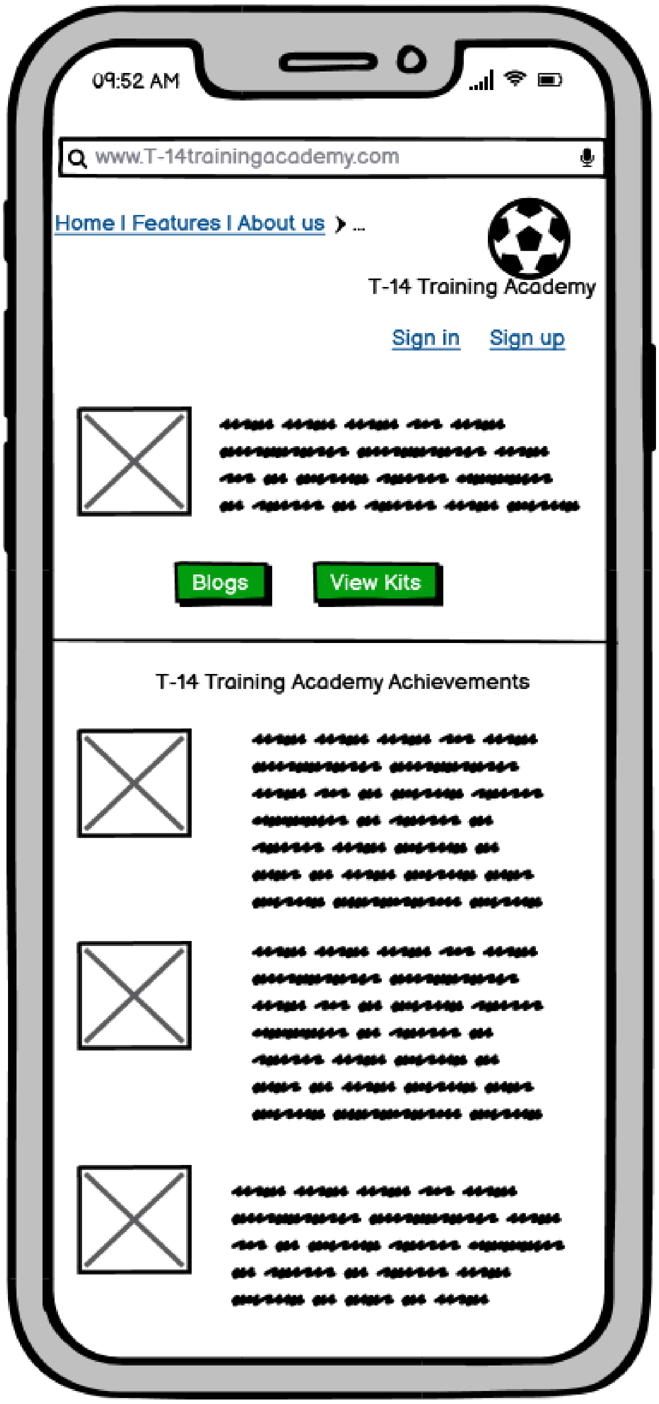

09:52 AM
www.T-14trainingacademy.com
Home I Features I About us
T-14 Training Academy
Sign in
Sign up
Blogs
View Kits
T-14 Training Academy Achievements

Figure 17: Landingpage

4. View product

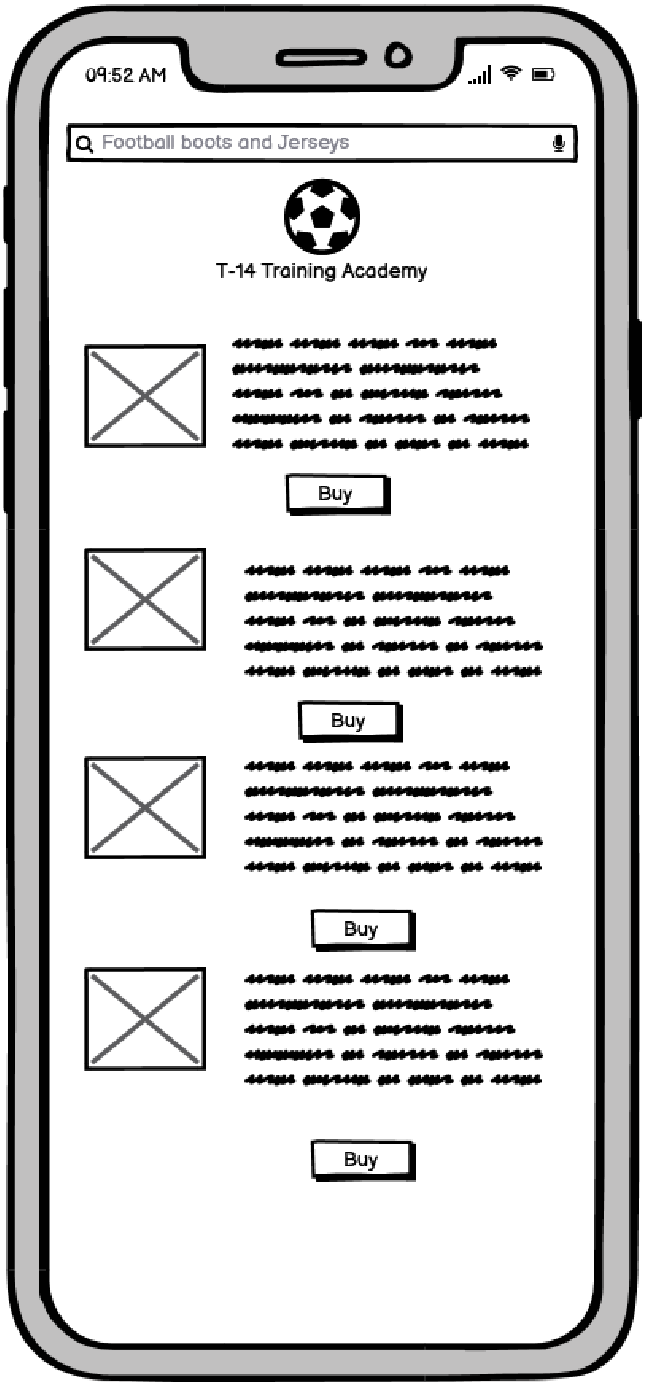


Figure 18: View product

5. Student Records

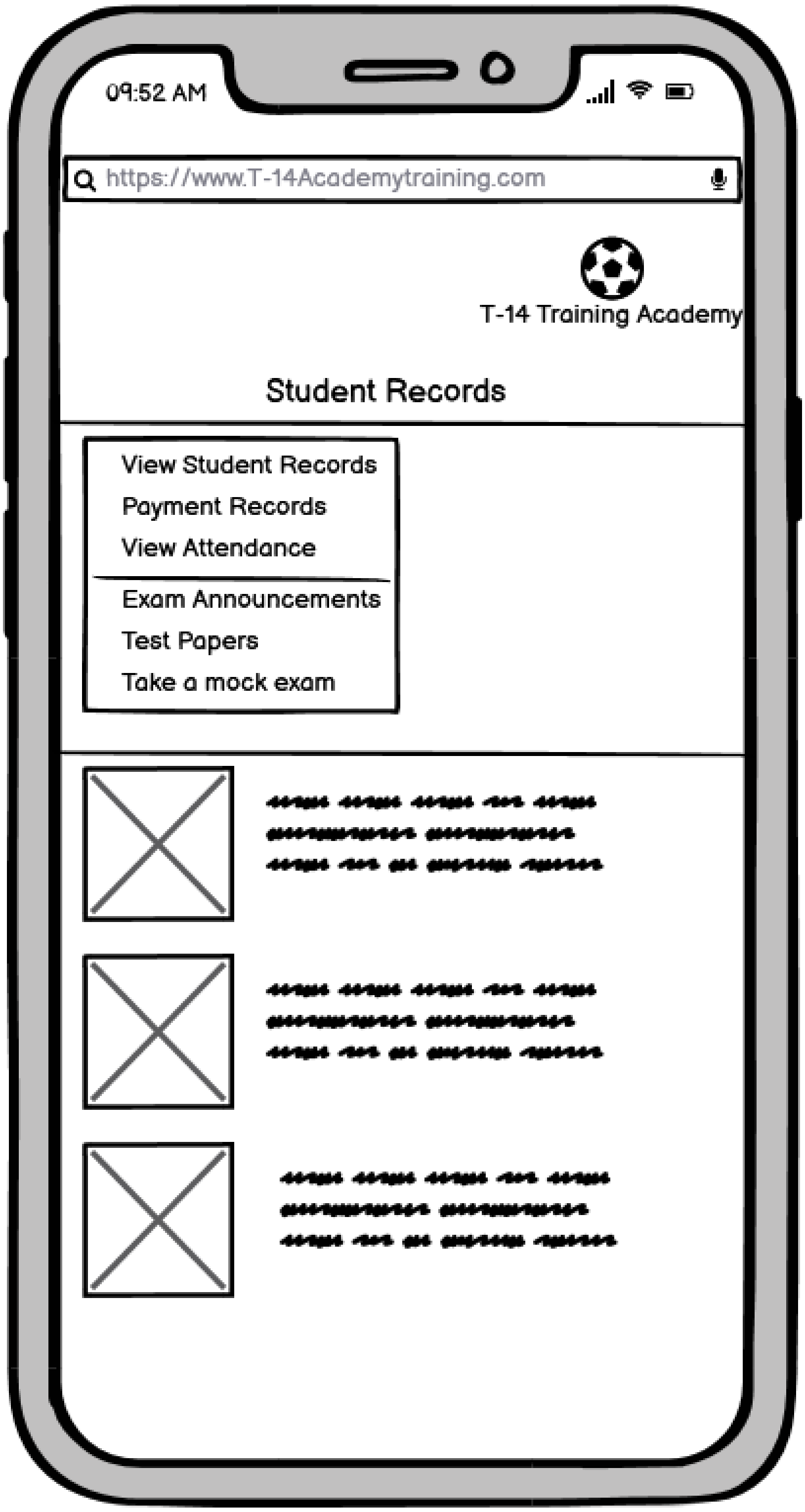
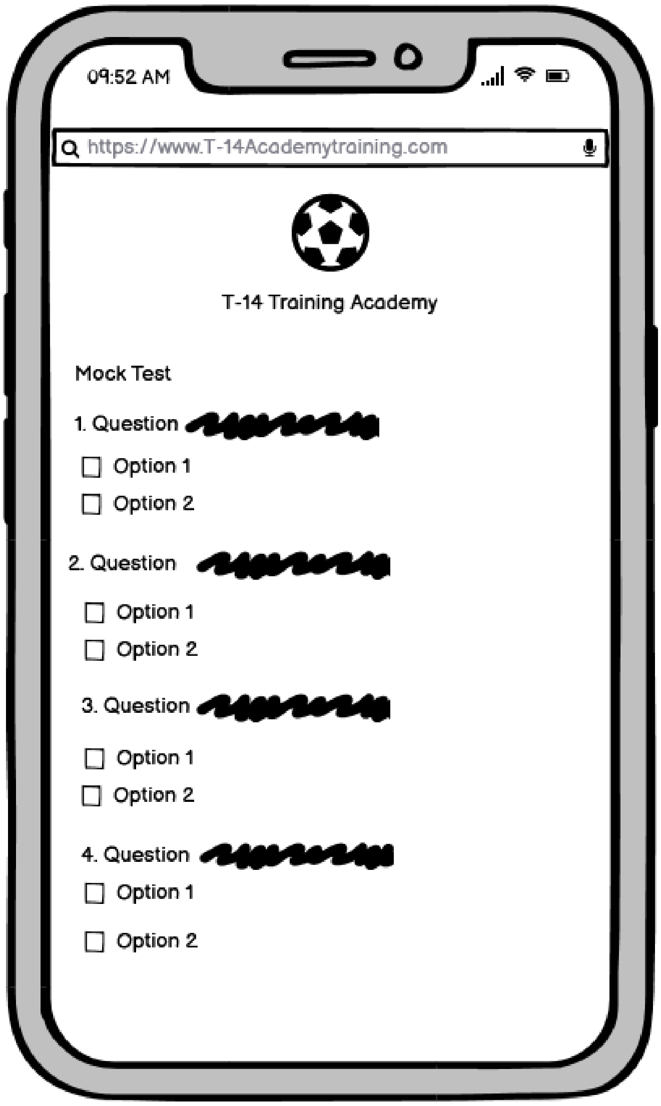


Figure 19: Student Records

6. Take mock test



09:52 AM

https://www.T-14Academytraining.com

T-14 Training Academy

Mock Test

1. Question
   - □ Option 1
   - □ Option 2

2. Question
   - □ Option 1
   - □ Option 2

3. Question
   - □ Option 1
   - □ Option 2

4. Question
   - □ Option 1
   - □ Option 2

Figure 20: Take mock test

7. Admin Register staff

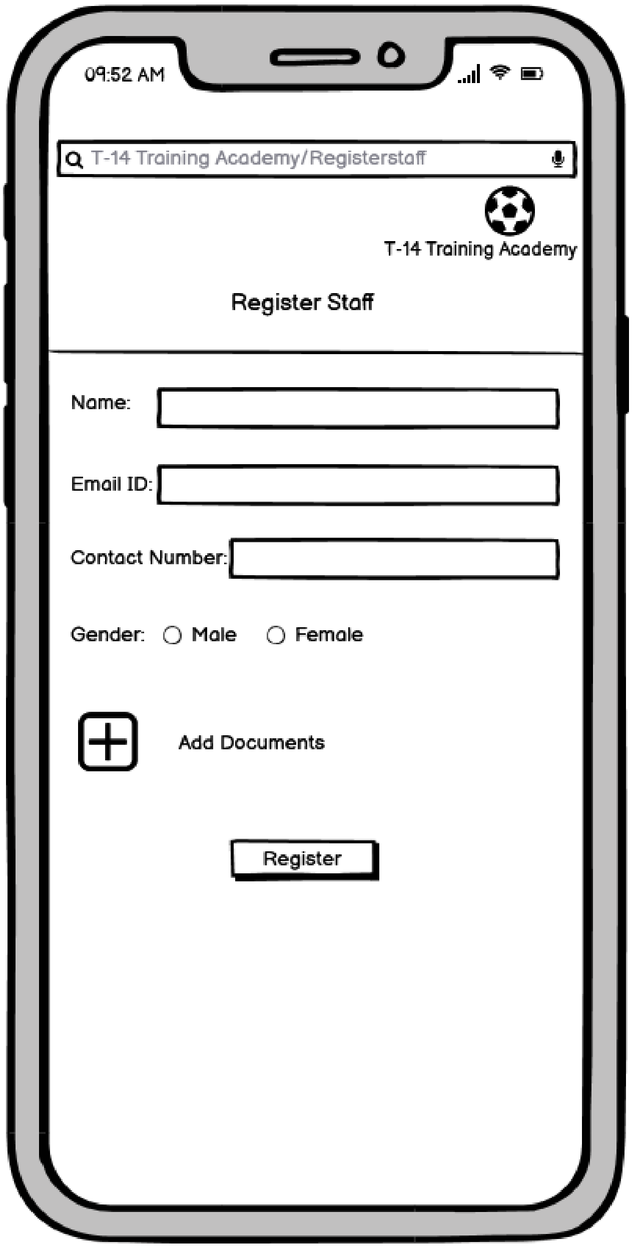


Figure 21: Register Staff

8. DE-Register Staff

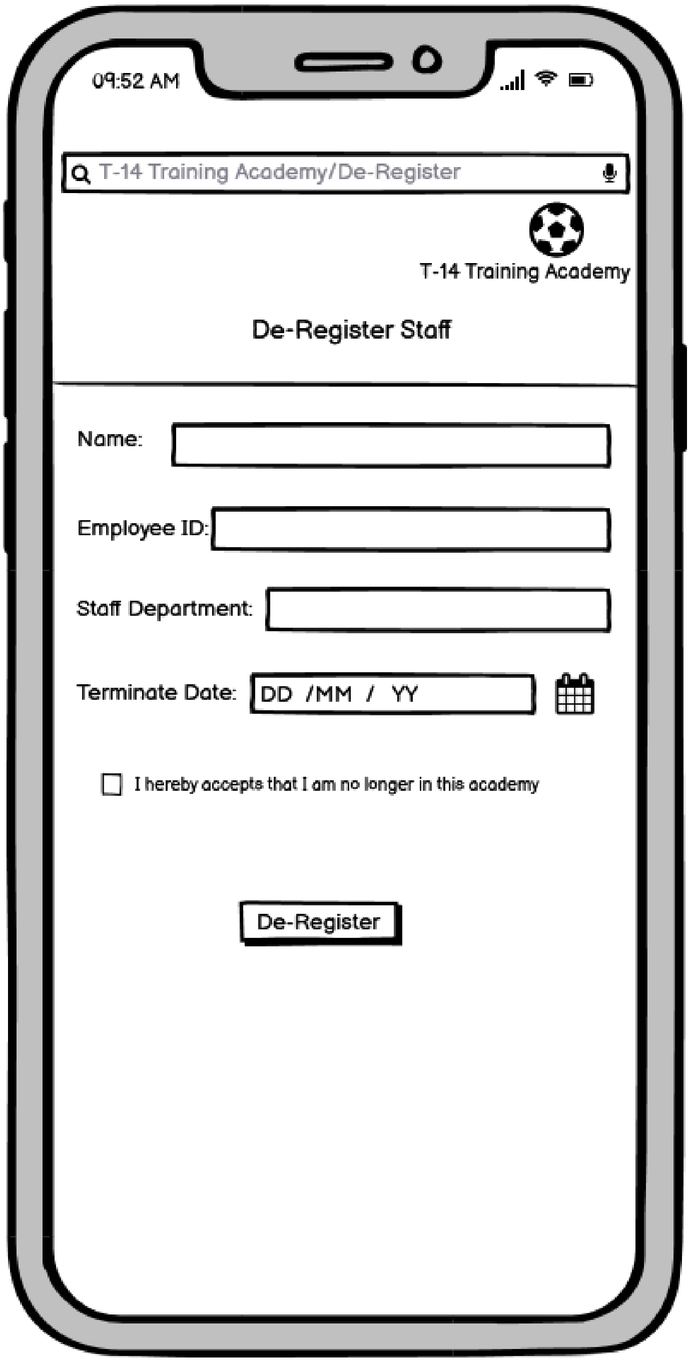
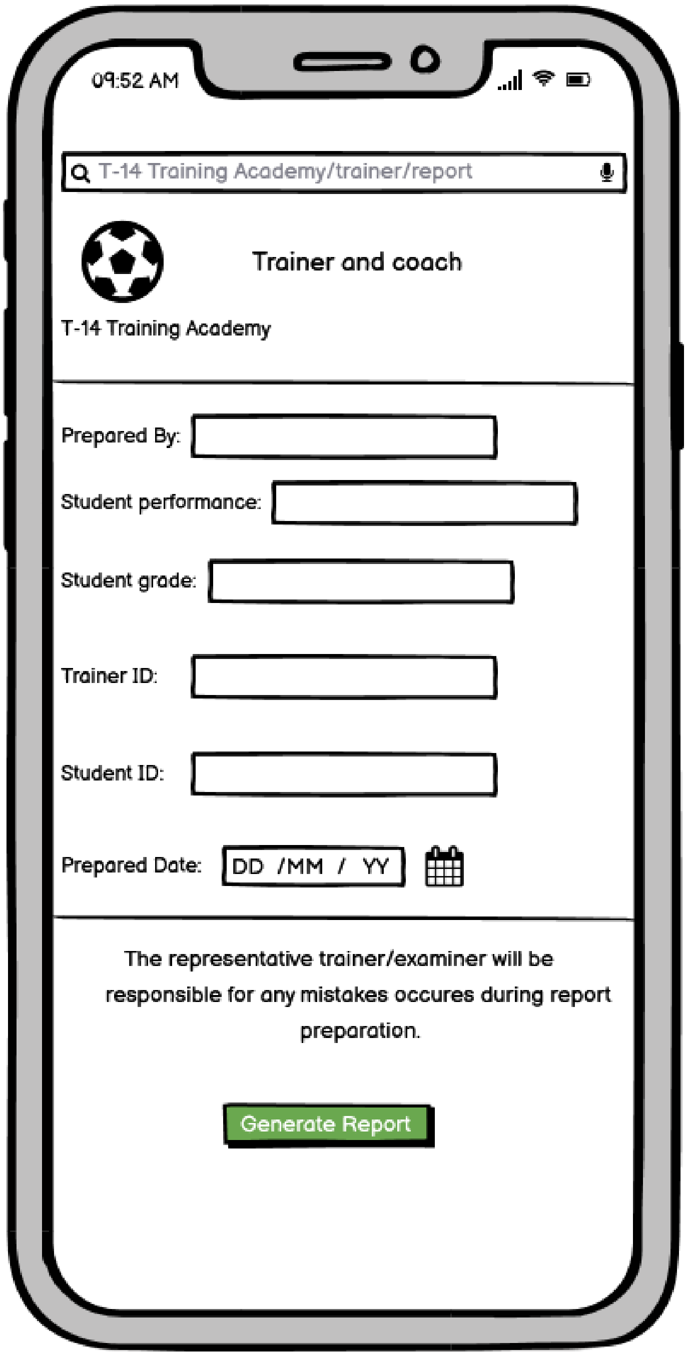


Figure 22: De-Register Staff

9. Generate report



Prepared By:

Student performance:

Student grade:

Trainer ID:

Student ID:

Prepared Date: DD /MM / YY

The representative trainer/examiner will be responsible for any mistakes occures during report preparation.

Figure 23: Generate report

9. Conclusion

The coursework 2 is finished effectively where I worked, arranged, and made logical synthesis of a digital framework growing of T-14 Training Academy. This coursework was conceivable by the guidelines of the trainers and furthermore through research and different sites.

This coursework helped in broadening my vision on the module and it likewise helped in creating extraordinary thoughts about software engineering, arranging, and scientific piece of the framework. This assignment has assembled a trust in me, and I am sure that this can help in my future. I acquired a great deal of information from this coursework

9.References

apm, 2022. *What is a Gantt chart?.* [Online]
Available at: https://www.apm.org.uk/resources/find-a-resource/gantt-chart/
[Accessed 05 May 2022].

creately, 2022. *Sequence Diagram Tutorial – Complete Guide with Examples.* [Online]
Available at: https://creately.com/blog/diagrams/sequence-diagram-tutorial/?msclkid=02c228f3ce7511ec95f786fe9b98a186
[Accessed 08 May 2022].

javaTpoint, 2022. *Use-Case Model.* [Online]
Available at: https://www.javatpoint.com/use-case-model
[Accessed 05 May 2022].

tutorialspoint, 2022. *UML - Class Diagram.* [Online]
Available at:
https://www.tutorialspoint.com/uml/uml_class_diagram.htm?msclkid=6583d498ce7611ec99d632435f616d5f
[Accessed 08 May 2022].

visual-paradigm, 2022. *What is Class Diagram?.* [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/?msclkid=65836e5dce7611ecb8daf35a28cde01b
[Accessed 08 May 2022].

visual-paradigm, 2022. *What is Communication Diagram?.* [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-communication-diagram/
[Accessed 08 May 2022].