Software Engineering (Object-Oriented Analysis and Design of the Proposed System)

Report By: Sayush Khadka

Table of Contents

Table of tables:

Table of figures:

1. Introduction:

The coursework is an online system designed for an organization named T-14 Training Academy. Object Oriented Analysis and Design of the proposed system are carried out.

Object-Oriented Analysis (OOA) is a method for learning about and modelling the demands of a system by a development team. In OOA, requirements are organized as objects. It brings all processes and data together. In certain circumstances, such as traditional structural analysis, both the method and the data are assessed separately. They employ flow charts/structure charts to represent processes and ER diagrams to represent data (Alam, 2019).

In OOA, however, various sophisticated models are used. Two of the most common models used in OOA are use cases and object models. Visual representations or overviews of the system's standard domain functions are called use cases. Object models describe the names, class connections, behaviours, and properties of important objects. User interface prototypes may also be created to enhance comprehension (Alam, 2019).

In Object-Oriented Analysis (OOA), the issue domain is the initial stage (the area of expertise or application that needs to analyse to solve a problem). Its purpose is to develop a conceptual model of the data in the region being studied. It might be in the form of a formal document, a written needs statement, interviews with stakeholders and other interested parties, or some other method. Object-oriented analysis will result in a conceptual model that outlines what the system is intended to do in terms of functionality (Alam, 2019).

Objective:

a) Register membership

b) Design exam test papers and practice test papers

c) Enroll staff members

d) Purchase football kits

e) Report preparation

f) Post exams notices and announcements

g) Take a mock exam

2. Gantt Chart:

A Gantt chart is a frequently used graphical depiction of a project timetable. It's a type of bar chart that shows the beginning and ending dates of project elements including resources, planning, and dependencies (Grant, 2021).

Henry Gantt (1861-1919), an American mechanical engineer, invented the Gantt chart (Grant, 2021).

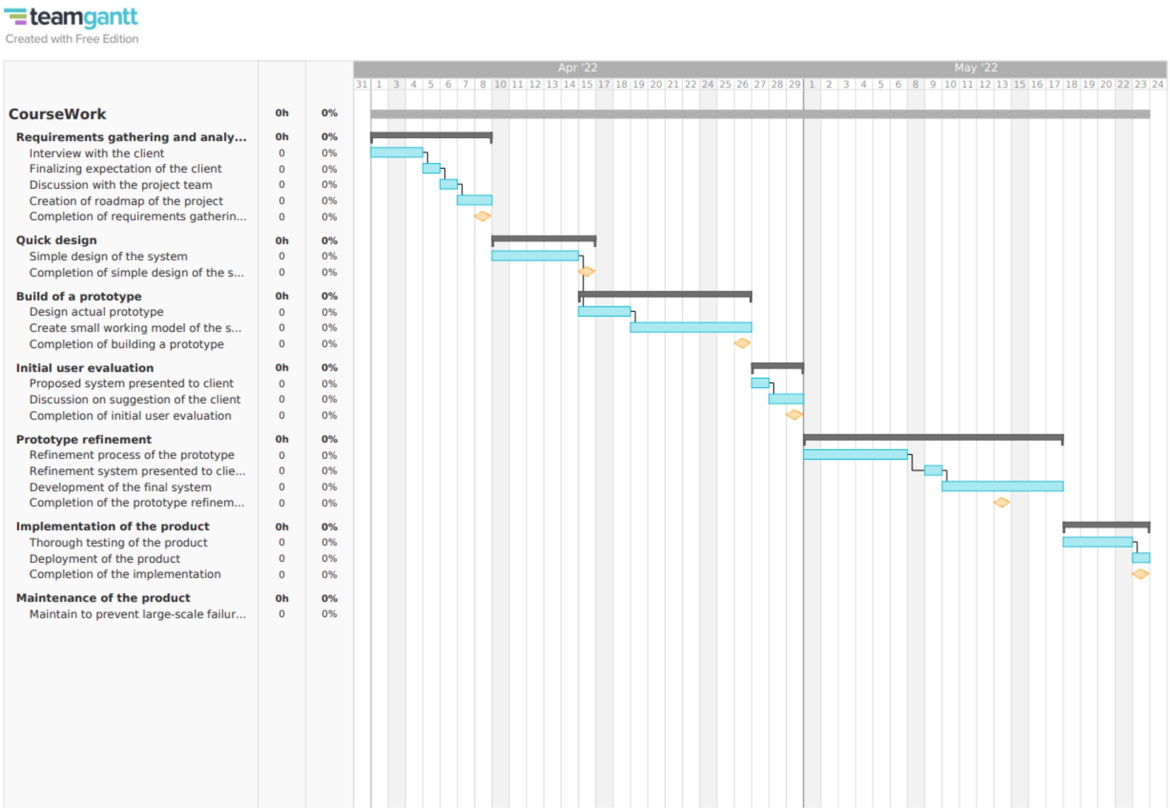
The Gantt chart is the most used in project management. These diagrams are helpful for planning a project and defining the sequence in which tasks must be done. The graph is frequently shown as a horizontal bar graph (Grant, 2021).

Horizontal bars of varying lengths depict the project timetable, which may include work sequences, duration, and start and completion dates for each job. The horizontal bar also shows how much of a task is left to do (Grant, 2021).

A Gantt chart is used to plan, organize, and track certain tasks and resources for a project. The graph shows the project timeline, which shows the work that has been scheduled and completed through time. The Gantt chart aids project managers in communicating project status and plans while also keeping the project on track (Grant, 2021).

The length of the bar indicates how long it takes to perform a task. The project tasks have been depicted by the vertical axis (Grant, 2021).

In the development of this system prototype methodology has been followed as the prototyping model is a systems development method in which a prototype is built, tested, and tweaked until an acceptable result is achieved, from which the whole system or product may be created. This strategy works effectively when not all of the project requirements are known in advance. It is an iterative, trial-and-error process for both developers and users (Lewis, 2019).



CourseWork	**0h**	**0%**
Requirements gathering and analy...	**0h**	**0%**
Interview with the client	0	0%
Finalizing expectation of the client	0	0%
Discussion with the project team	0	0%
Creation of roadmap of the project	0	0%
Completion of requirements gatherin...	0	0%
Quick design	**0h**	**0%**
Simple design of the system	0	0%
Completion of simple design of the s...	0	0%
Build of a prototype	**0h**	**0%**
Design actual prototype	0	0%
Create small working model of the s...	0	0%
Completion of building a prototype	0	0%
Initial user evaluation	**0h**	**0%**
Proposed system presented to client	0	0%
Discussion on suggestion of the client	0	0%
Completion of initial user evaluation	0	0%
Prototype refinement	**0h**	**0%**
Refinement process of the prototype	0	0%
Refinement system presented to clie...	0	0%
Development of the final system	0	0%
Completion of the prototype refinem...	0	0%
Implementation of the product	**0h**	**0%**
Thorough testing of the product	0	0%
Deployment of the product	0	0%
Completion of the implementation	0	0%
Maintenance of the product	**0h**	**0%**
Maintain to prevent large-scale failur...	0	0%

Figure 1 Gantt Chart

Exact time taken to complete the system with proper testing and deployment is 45 days.

3. Use Case Model:

A use-case model describes how individuals engage with a technology to address a problem. Therefore, the use case model outlines the user's purpose, the system's interactions with the user, and the system's behaviour that is required to accomplish these goals (javaTpoint, n.d.).

3.1 Use Case Diagram:

A use case diagram is a visual depiction of a system's and its users' characteristics. It's frequently portrayed as a graphic picture of how various pieces of a system interact. Use case diagrams describe the activities that take place in a system and how they flow, but they don't specify how those events are carried out (Contributor, 2020).

Use case diagram working:

a) The boundary between the interest system and the rest of the world (Contributor, 2020).

b) The roles of the actors, who are often people who participate in the system, define them (Contributor, 2020).

c) The use cases or the specific roles that actors within and outside the system play (Contributor, 2020).

d) Interactions between and among participants, as well as use cases (Contributor, 2020).

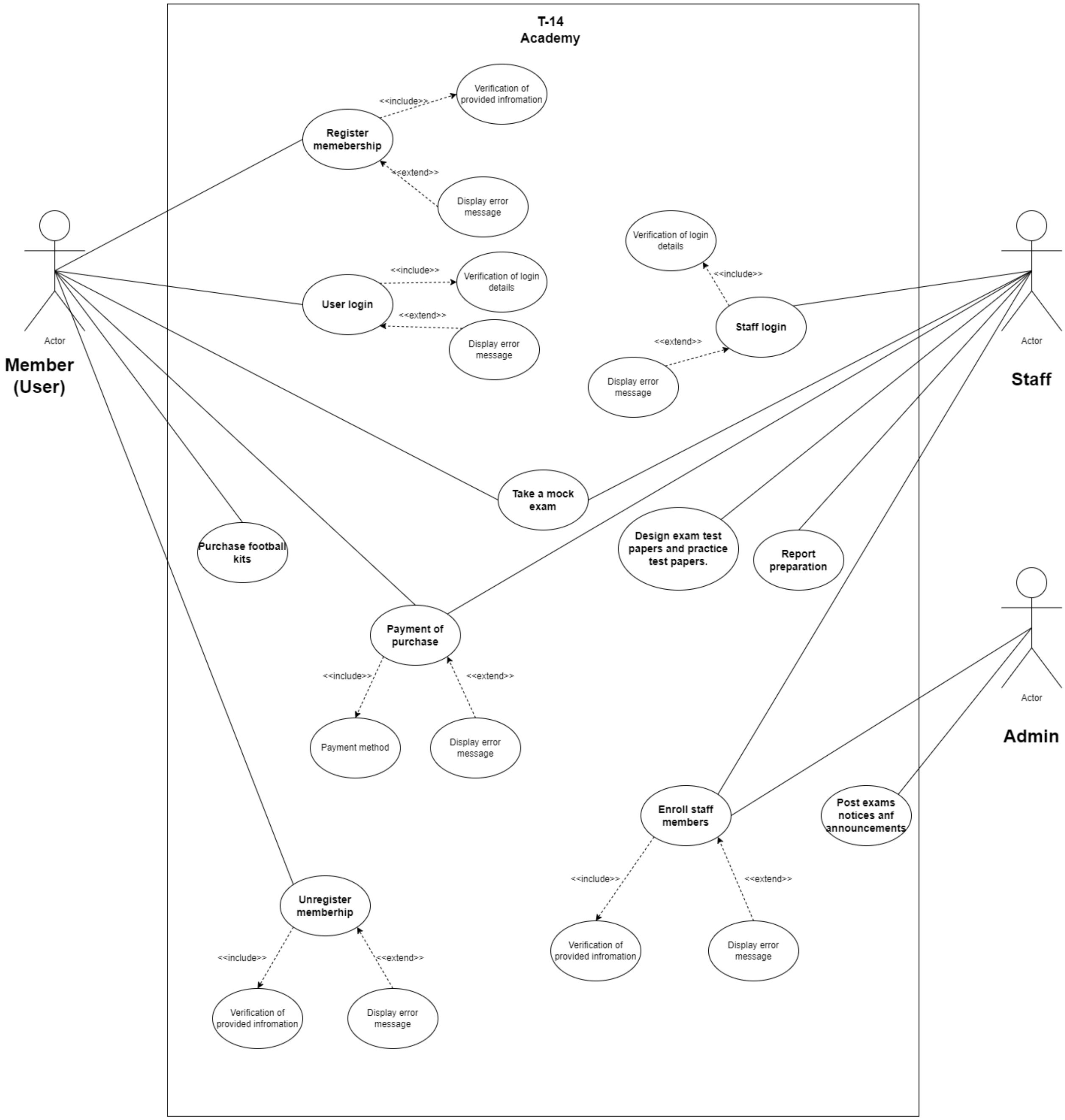


Figure 2 Use case diagram

3.2 High Level Use Case Description:

3.2.1 Use case: Register membership

Use Case:	Register membership
Actors:	New Member (Student)
Description:	A new member provides personal details to the system as a student to become an online member of the academy.

Table 1 Use case: Register Membership

3.2.2 Use Case: User login

Use Case:	User login
Actors:	Member (User)
Description:	After the registration of the member, he/she is provided with the login username and password. The member can login to the system using the correct username and password.

Table 2 Use Case: User login

3.2.3 Use case: Staff login

Use Case:	Staff login
Actors:	Staff
Description:	The staffs are provided with their specific login username and password to have staff privilege. The staff can login to the system using their provided username and password respectively.

Table 3 Use case: Staff login

3.2.4 Use case: Design exam test papers and practice test papers

Use Case:	Design exam test papers and practice test papers.
Actors:	Staff (Coaches/Trainers)
Description:	The staff designs the exam test papers and practice test papers (MCQ, Subjective) in the system itself for the member (students).

Table 4 Use case: Design exam test papers and practice test papers

3.2.5 Use case: Enroll staff members

Use Case:	Enroll staff members
Actors:	Staff, Admin
Description:	The enrolling staff provides the personal and professional details required by the organization as to become the staff of the organization. The information is then verified by the system as well as admin. The admin finally enrolls the staff if the enrolling staff meets the requirement of the organization.

Table 5 Use case: Enroll staff members

3.2.6 Use case: Purchase football kits

Use Case:	Purchase football kits
Actors:	Member
Description:	For members of the academy, there is a special discount for buying kits (Jerseys, Socks, Boots, training tracksuits etc.) for buying the items listed in the system especially the Jersey of Professional footballer clubs. Member has already logged in to the system.

Table 6 Use case: Purchase football kits

3.2.7 Use case: Payment of purchase

Use Case:	Payment of purchase
Actors:	Member, Staff
Description:	The member can pay for the purchase of the goods through different payment method (banking transaction/ mobile banking/cash). Purchase transaction is monitored by the staff.

Table 7 Use case: Payment of purchase

3.2.8 Use case: Report preparation

Use Case:	Report preparation
Actors:	Staff (Trainer/Examiner)
Description:	The trainer/examiner can enter the performance score and generate the reports of members regarding their performance in the written and physical tests.

Table 8 Report preparation

3.2.9 Use case: Post exams notices and announcements

Use Case:	Post exams notices and announcements
Actors:	Admin
Description:	The admin can post notices, results and other announcements related with the organization.

Table 9 Use case: Post exams notices and announcements

3.2.10 Use case: Take a mock exam

Use Case:	Take a mock exam
Actors:	Member (intermediate training enrollment), Staff
Description:	The member enrolling to intermediate training need to watch the football related videos and give mock exams and see results. These exams are hosted and marked by the staff.

Table 10 Use case: Take a mock exam

3.2.11 Use case: Unregister membership

Use Case:	Unregister membership
Actors:	Member
Description:	The members can also unregister themselves from the organization system. They are required to provide their correct respective username and password to unregister.

Table 11 Use case: Unregister membership

3.3 Expanded Use Case Description:

3.3.1 Use Case: Register membership

Use Case: Register membership

Actor: Member (Student)

Description: A new member provides personal details to the system as a student to become an online member of the academy.

Typical Course of event:

Actor action:	System response:
1. The member(student) opens the system.	2. System interface gets displayed.
3. The member gets list of course (basic or intermediate).	4. System displays the list of course (basic or intermediate).
5. The member fills in the required information asked by the system.	6. Systems takes the input and validates it.
7. The member chooses the course (basic or intermediate)	8. System takes the input.
9. The member requests to register clicking on the register button.	10. If the entered information is correct after the verification of the information, system registers the member and provides with username and password for the registered member.

Table 12 Use case (Expanded): Register membership

Alternate courses:

Line 7 and 8: If the user entered information is incorrect or the verification process fails, the system displays error message allowing user to enter valid and correct information and the use case ends.

3.3.2 Use Case: Design exam test papers and practice test papers

Use case: Design exam test papers and practice test papers

Actor: Staff (Coaches/Trainers)

Description: The staff designs the exam test papers and practice test papers (MCQ, Subjective) in the system itself for the member (students).

Typical Course of event:

Actor action:	System response:
1. The staff logs in using their given staff username and password.	2. The system validates the given information.
	3. The system allows the person to log in only if the entered information is valid.
4. The staff has access to the staffs' privileges or staff mode.	5. The system provides the staff with different staff features.
6. The staff designs exam test papers and practice test papers in the system itself.	7. The system allows the staff to design exam test papers and practice test papers in the system itself.

Table 13 Use Case: Design exam test papers and practice test papers

Alternate courses:

Line 3: If the entered log in information by the staff is not validate, the system displays an error message.

Line 4: The staff does not get logged in. The staff is again asked to enter valid information.

4. Communication diagram:

UML Communication Diagrams, also known as collaboration diagrams, are a type of behavioural diagram that shows how objects in software or system communicate. This diagram highlights the communications sent and received between objects. Communication diagrams are the ideal choice when one use case has several circumstances to portray (EdrawMax, n.d.).

4.1 Collaboration diagram:

The relationship between the objects in a system is depicted in a collaboration diagram. Both the sequence and collaboration diagrams show the same information but in different ways. Because it is based on object-oriented programming, it depicts the architecture of the object existing in the system rather than the flow of messages. A product has a variety of qualities. Multiple elements of the system are interconnected. The collaboration diagram, also known as a communication diagram, is used to show how an item in a system works (javaTpoint, n.d.).

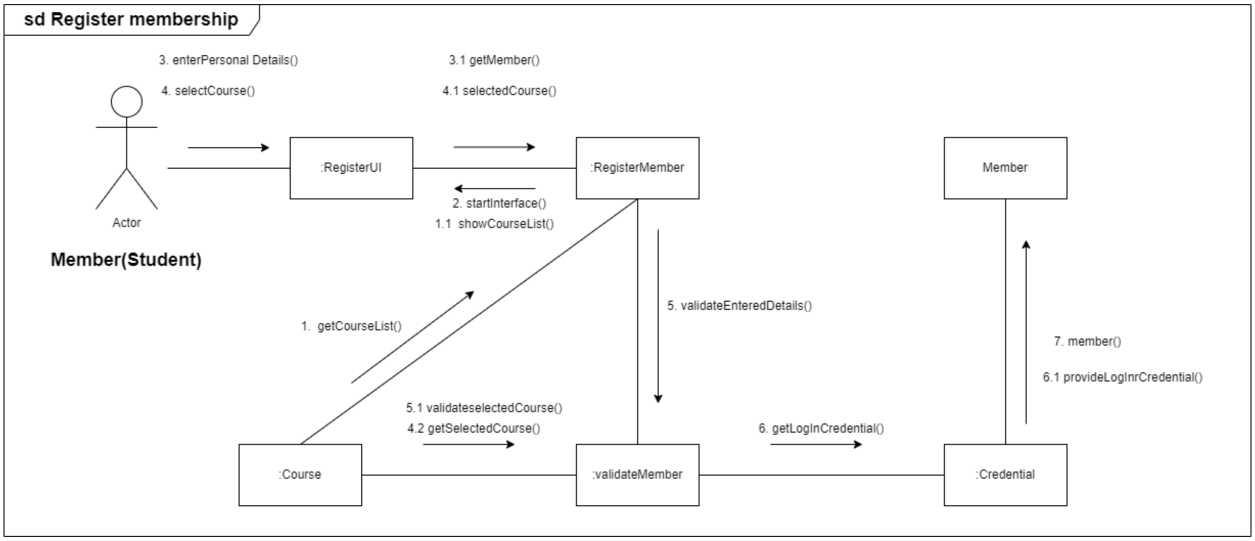


Figure 3 Collaboration diagram

4.2 Sequence diagram:

Interactions between classes are depicted as a series of messages exchanged over time in sequence diagrams. They're also known as event diagrams. A sequence diagram can assist you in visualizing and testing various runtime scenarios. During the modelling phase, these can help forecast how a system will act and identify responsibilities that a class may require (smartdraw, n.d.).

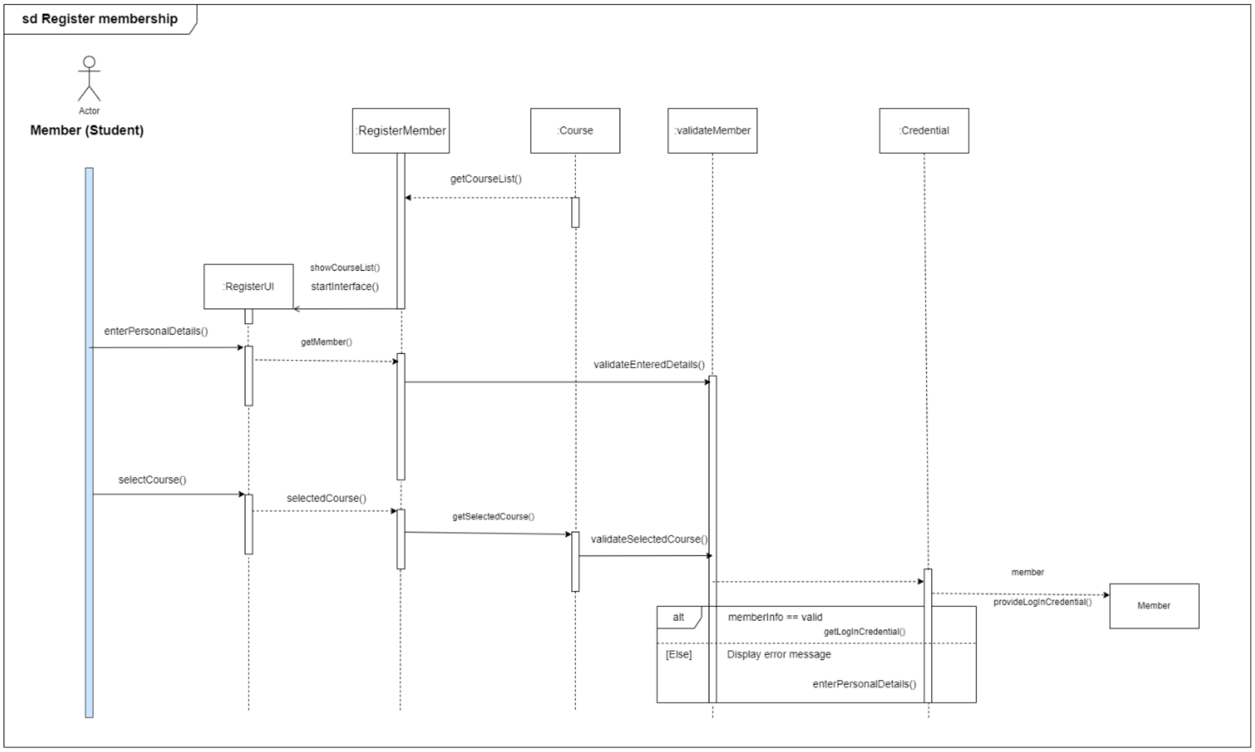


Figure 4 Sequence diagram

5. Class diagram:

A class diagram is a type of diagram that shows the overall structure and overview of a system in terms of classes, attributes, and methods, as well as the relationships between them. The unified modelling language includes it (UML) (techopedia, n.d.).

It's used to display and build a functional diagram of system classes, as well as serve as a system development resource throughout the software development life cycle (techopedia, n.d.).

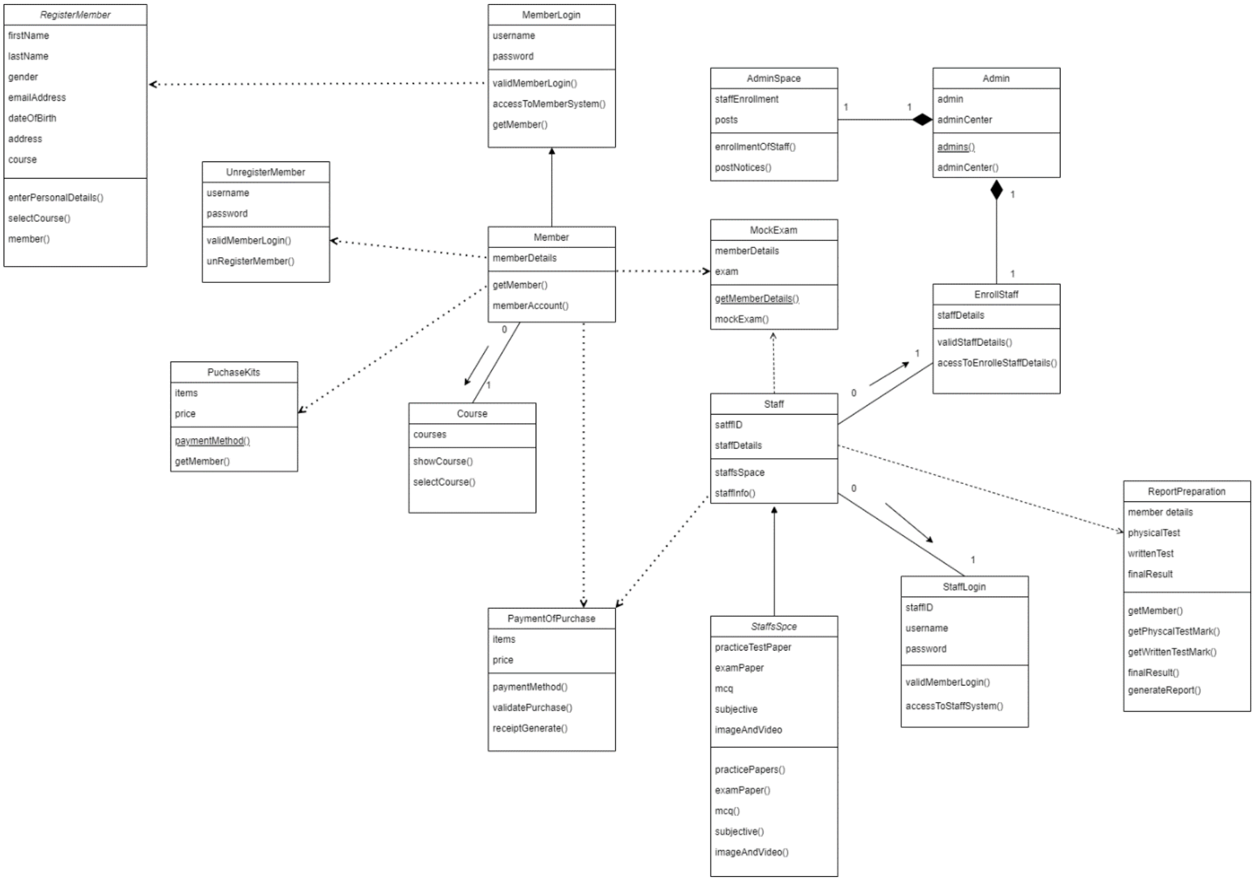


Figure 5 Class diagram

6. Further Development process:

This system has been designed for an organization named T-14 Training Academy. Object-Oriented Analysis and Design of the proposed system are carried out. Different analyses and design diagrams have been completed.

Architecture Pattern (Event-driven Architecture):

So, for further design development of the system Event-driven Architecture Pattern (Software Architecture) is proposed and is used. The majority of programs spend the majority of their time waiting for something to happen. This is particularly true for computers that interact directly with humans, but it also happens often in other areas like networks. Data must be processed on occasion, but not all the time. The event-driven architecture makes this possible by creating a central unit that collects all data and distributes it to the many modules that handle each category. This handoff is said to trigger an "event," which is then delegated to the type's code (Wayner, n.d.).

Writing small modules that react to events like mouse clicks or keystrokes is part of the JavaScript development process. All input is orchestrated by the browser, which guarantees that only the right code sees the right events. There are many different types of events in the browser, but the modules only interact with the ones that interest them. A layered design, on the other hand, requires all data to transit through all layers (Wayner, n.d.).

Moreover, event-driven architecture pattern is chosen for this system as they are:

a) Adapt well to difficult, frequently chaotic environments (Wayner, n.d.).

b) Event-driven architecture easily scales (Wayner, n.d.).

c) When additional event kinds occur, they are simply expanded (Wayner, n.d.).

Usage:

a) When just a few modules interact with data blocks (Dhaduk, 2020).

b) Assists with user interface design (Dhaduk, 2020).

Methodology (Prototype methodology or prototype model):

The development plan and process took following the Prototype methodology. The prototyping model is a systems development method in which a prototype is built, tested, and adjusted until an acceptable result is achieved, from which the whole system or product may be created. This strategy works effectively when not all of the project requirements are known in advance. It is an iterative, trial-and-error process for both developers and users (Lewis, 2019).

Why the prototype model has been used in this system:

a) Customers get an early say in the product, which boosts customer satisfaction (Lewis, 2019).

b) It places a strong emphasis on teamwork and adaptable design (Lewis, 2019).

c) Errors and missing features are promptly identified (Lewis, 2019).

d) In the future, prototypes may be used in more complex tasks (Lewis, 2019).

e) Users have a better understanding of the product's capabilities (Lewis, 2019).

f) Faster customer feedback allows us to have a better picture of what they require (Lewis, 2019).

Prototyping Model Phases (SDLC phases):

a) Requirements gathering and analysis: The first step in prototyping is to do a requirement analysis. At this stage, the system's needs are thoroughly established. Users of the system are questioned throughout the process to discover what they expect from it (Martin, 2022).

b) Quick design: The second phase is a preliminary design, sometimes known as a quick design. At this point, a rudimentary system design is created. It is not, however, a complete design. It provides a short overview of the system to the user. The rapid design facilitates prototype creation (Martin, 2022).

c) Build of a prototype: In this stage, the knowledge gathered during quick design is used to develop a genuine prototype. It's a simplified version of the required system (Martin, 2022).

d) Initial user evaluation: At this point, the proposed system is sent to the customer for an initial evaluation. It assists in assessing the strengths and limitations of the working model. Customer recommendations and reviews are collected and delivered to the developer (Martin, 2022).

e) Prototype refinement: If the user is dissatisfied with the current prototype, you must make changes in response to their suggestions and comments. This phase will last until all the user's requirements have been met. If the user is satisfied with the developed prototype, a final system is created based on the permitted final prototype (Martin, 2022).

f) Implementation of the product: When the final prototype is created, the system is extensively tested and put into production (Martin, 2022).

g) Maintenance of the product: To avoid downtime and large-scale failures, the system is subjected to routine maintenance (Martin, 2022).

Testing (White box testing):

White box testing is carried out for the overall system. White box testing is a sort of software analysis in which the basic structure, architecture, and code of a product are evaluated to improve design, usability, and security. Because code is visible to testers, white box testing is also known as clear box testing, open box testing, transparent box testing, code-based testing, and glass box testing (Hamilton, 2022).

It is one of two components of the software testing approach known as Box Testing. Black box testing, on the other hand, involves testing from the standpoint of an external or end-user. White box testing, on the other hand, focuses on the inner workings of an application and is used in software engineering for internal testing (Hamilton, 2022).

The term "White Box" was developed because of the see-through box concept. The term "clear box" or "White Box" refers to the capacity to see into the inner workings of software through its outside shell (or "box"). Similarly, the "black box" in "Black Box Testing" refers to the inability to examine the inner workings of the program, leaving just the end-user experience to be evaluated (Hamilton, 2022).

The following are some of the things/issues/functions that white box testing looks for:

a) Paths in the coding process that are broken or poorly organized (Hamilton, 2022).

b) Proper functioning of the inputs through the code (Hamilton, 2022).

c) Internal flaws in security (Hamilton, 2022).

d) Proper functioning of loops and conditions (Hamilton, 2022).

e) Each statement, object, and function are individually tested (Hamilton, 2022).

f) Test cases and their expected outputs (Hamilton, 2022).

The software is then deployed and released after the proper design and development as well as thorough testing.

The software is routinely maintained to avoid downtime and large-scale failures.

7. Prototype Development:

Prototype for “T-14 Training Academy”

7.1 Prototype: Home page

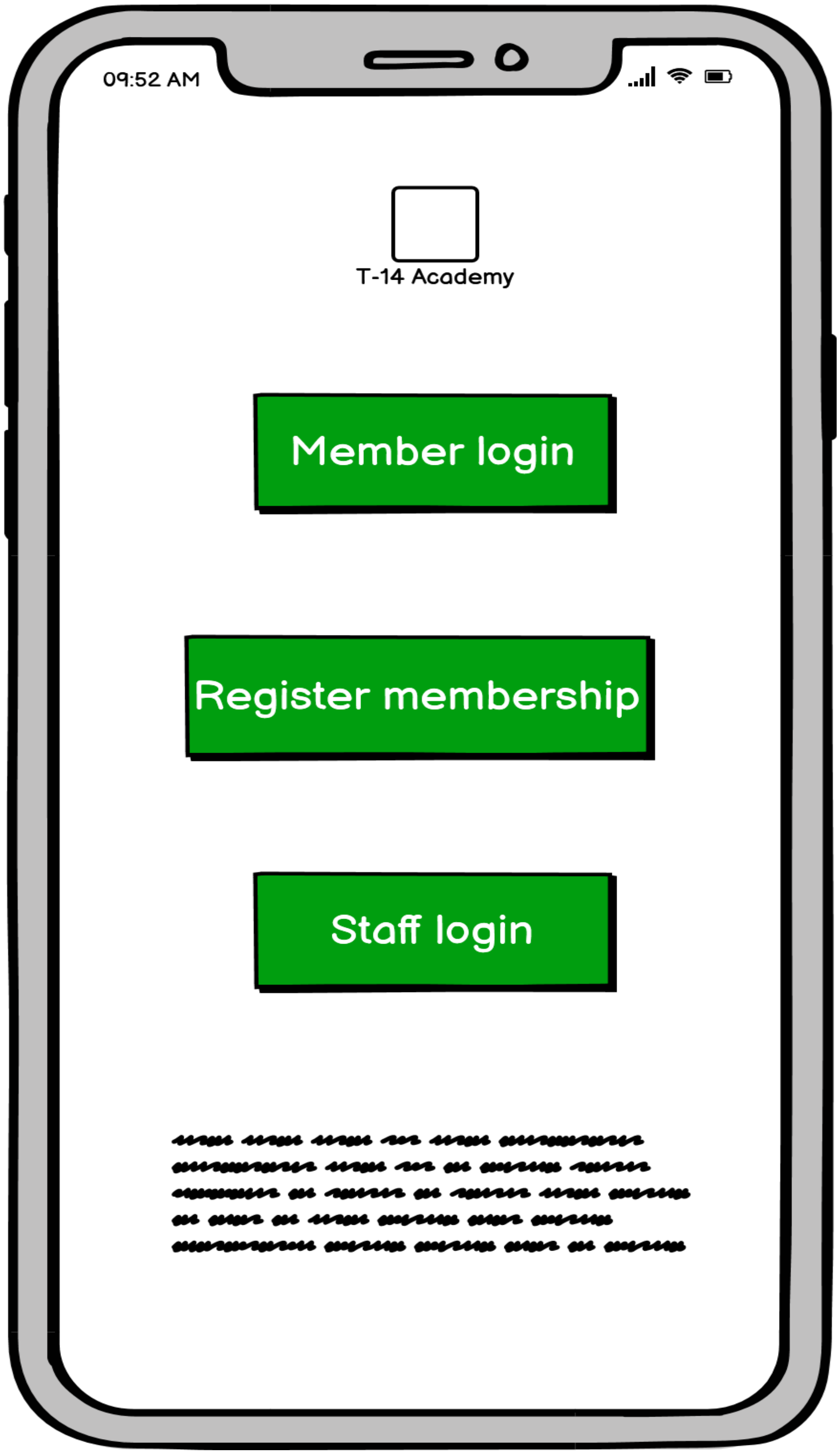


Figure 6 Prototype: Home page

7.2 Prototype: Member Login

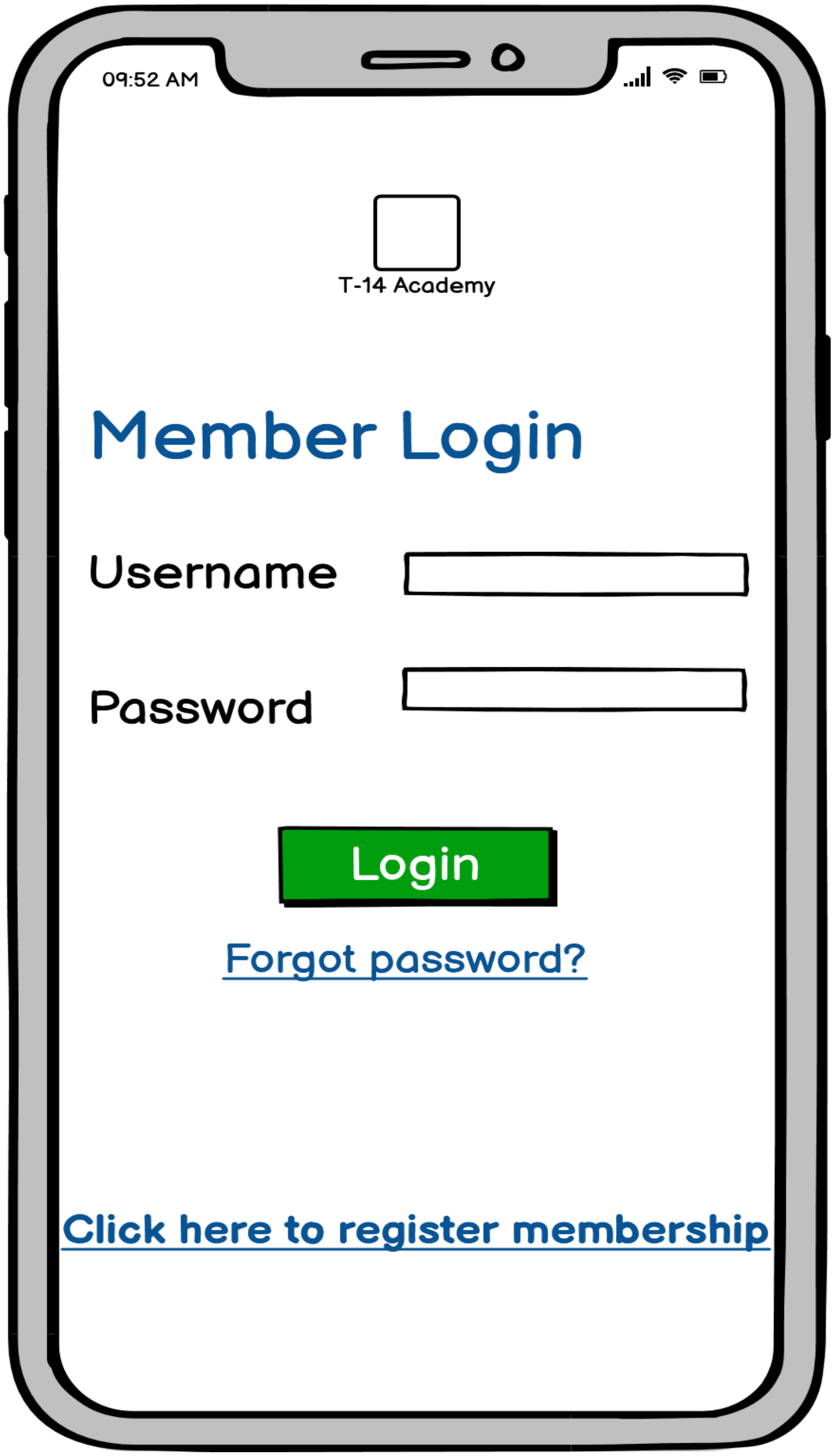
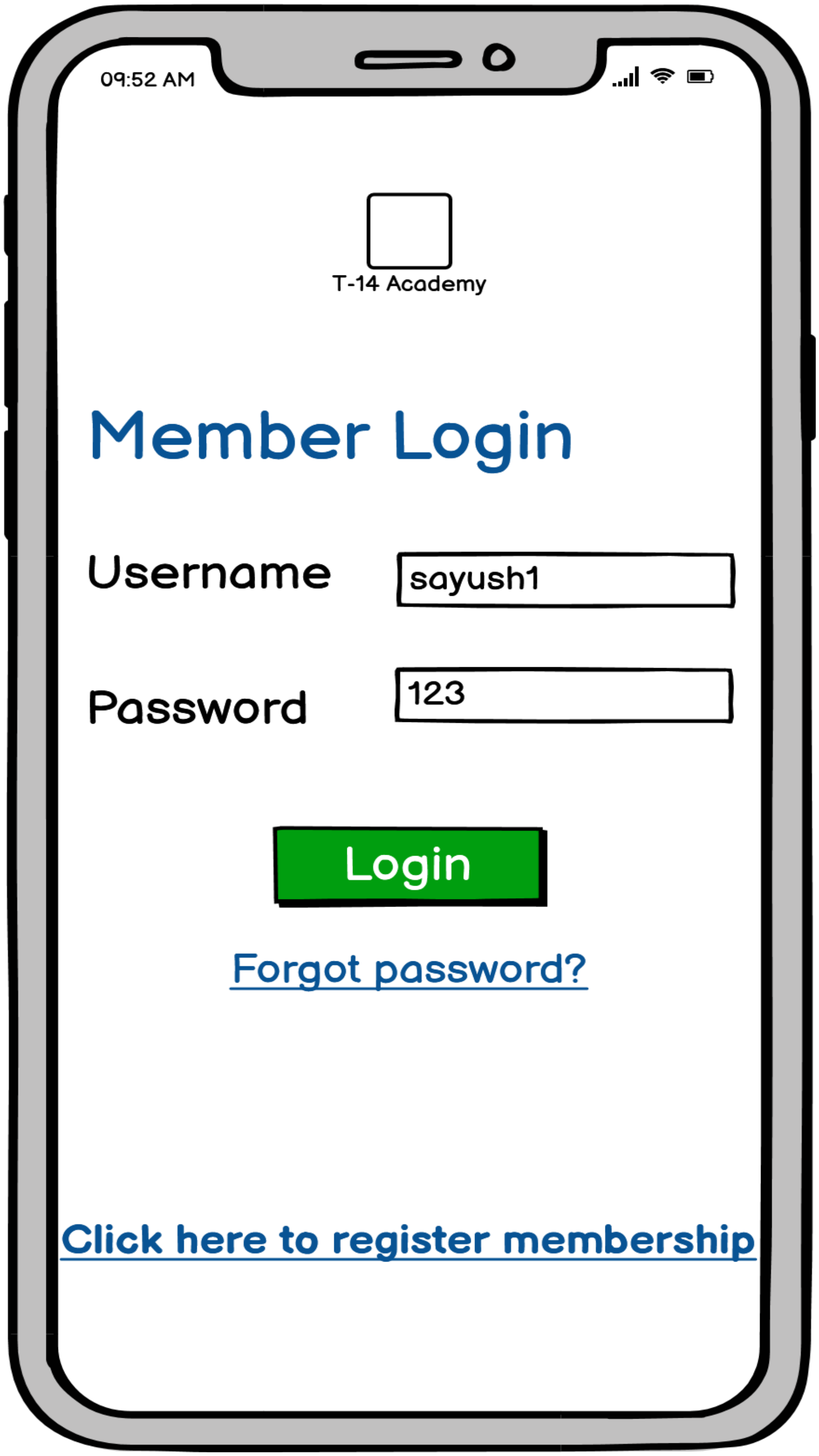


Figure 7 Prototype: Member Login

7.3 Prototype: Member login error details (Not registered)

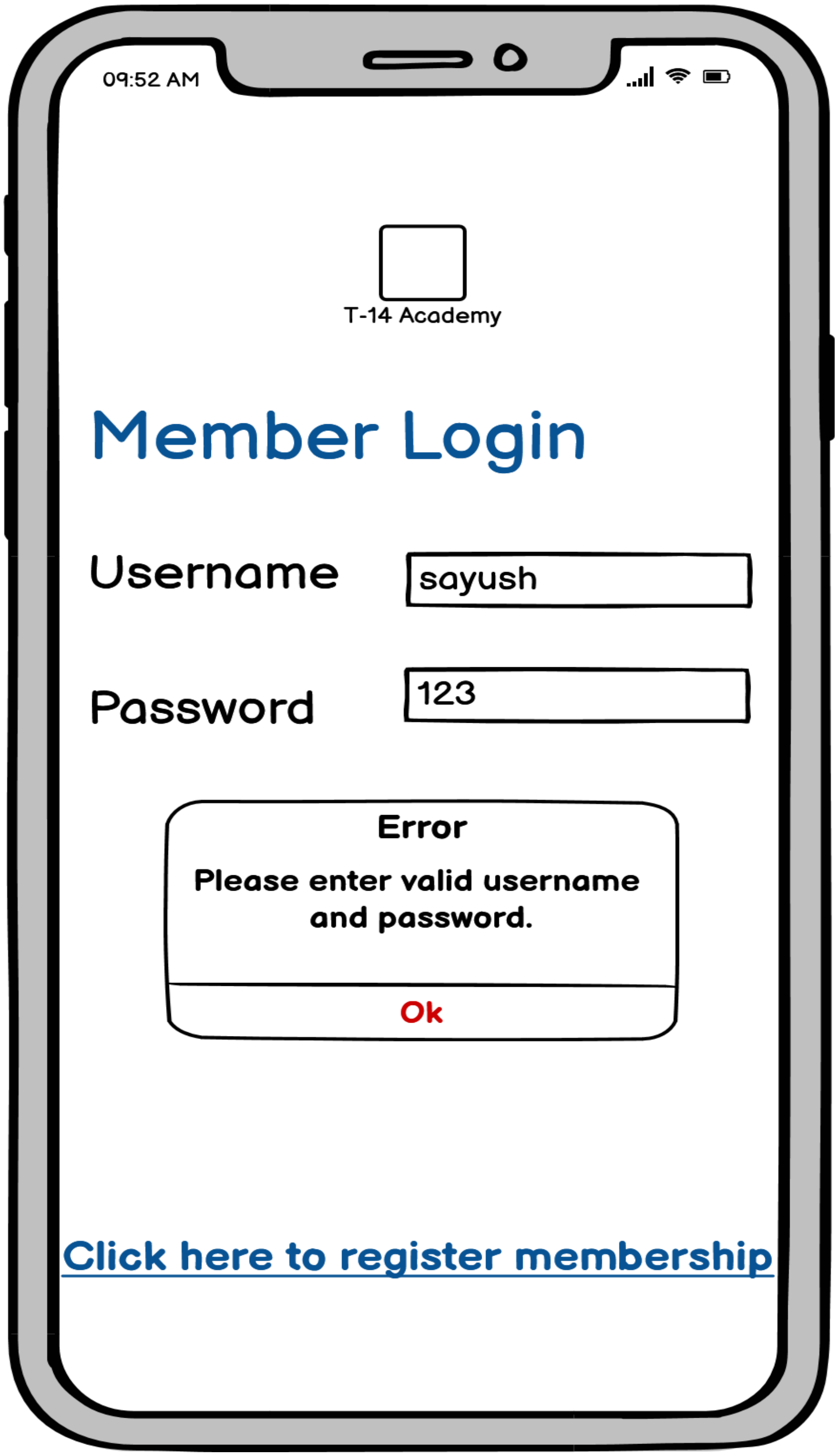


Username sayush1

Password 123

Figure 8 Prototype: Member login error details (Not registered)

7.4 Prototype: Member login error message



Username: sayush

Password: 123

Error

Please enter valid username and password.

Figure 9 Prototype: Member login error message

7.5 Prototype: Register membership (Page layout)

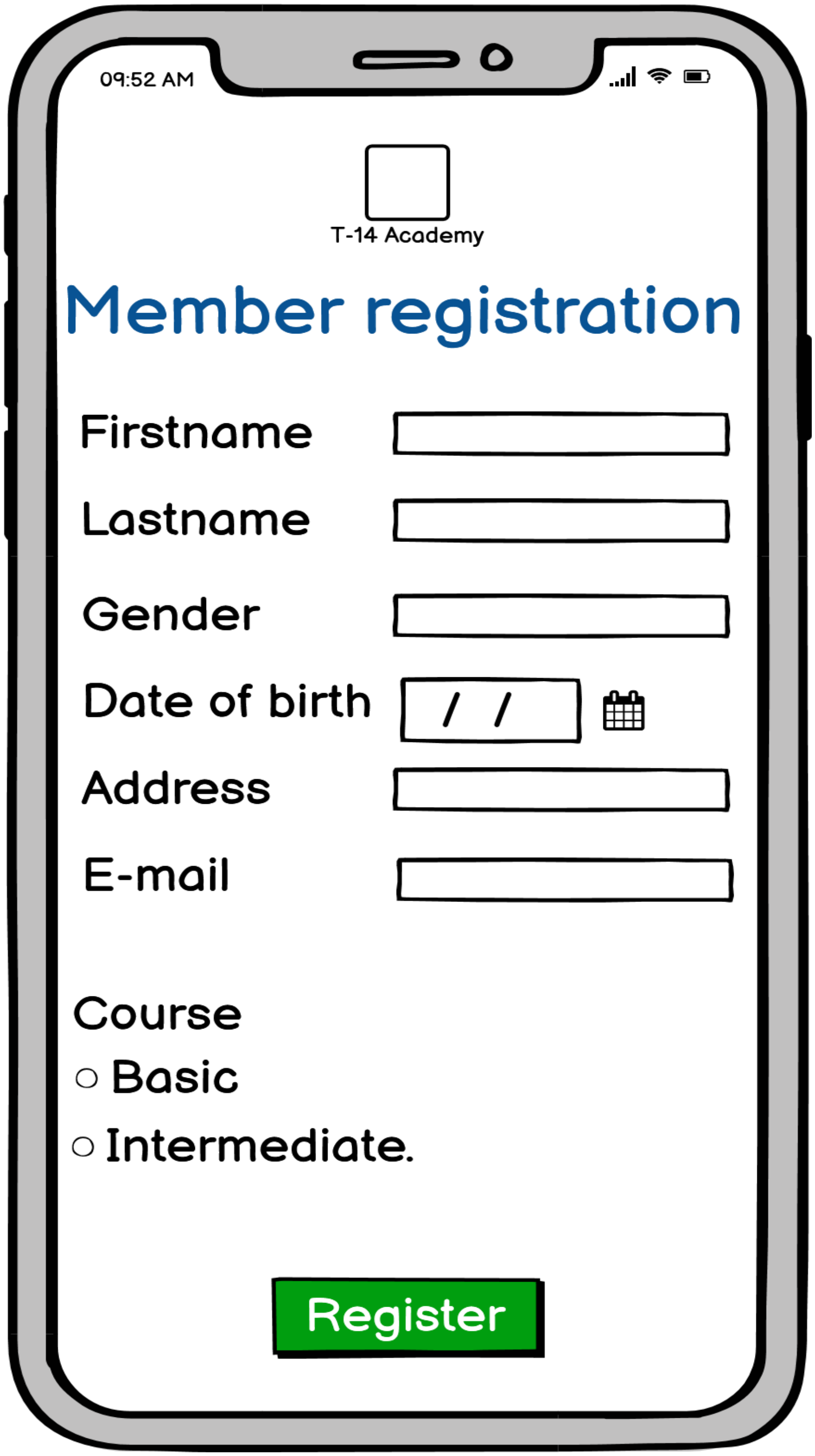
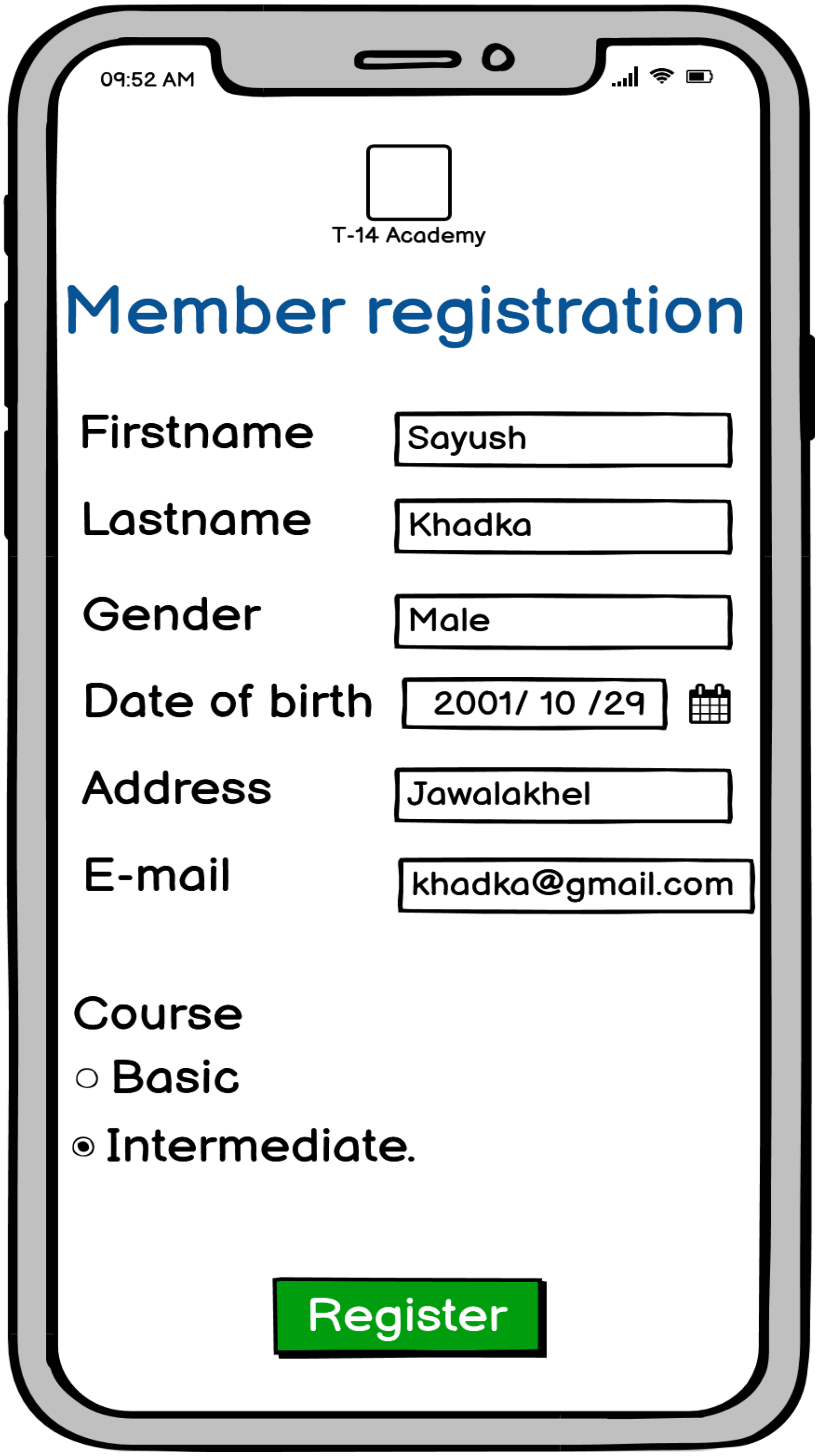


Figure 10 Prototype: Register membership (Page layout)

7.6 Prototype: Register membership (Valid registration information)



Firstname: Sayush

Lastname: Khadka

Gender: Male

Date of birth: 2001/ 10 /29

Address: Jawalakhel

E-mail: khadka@gmail.com

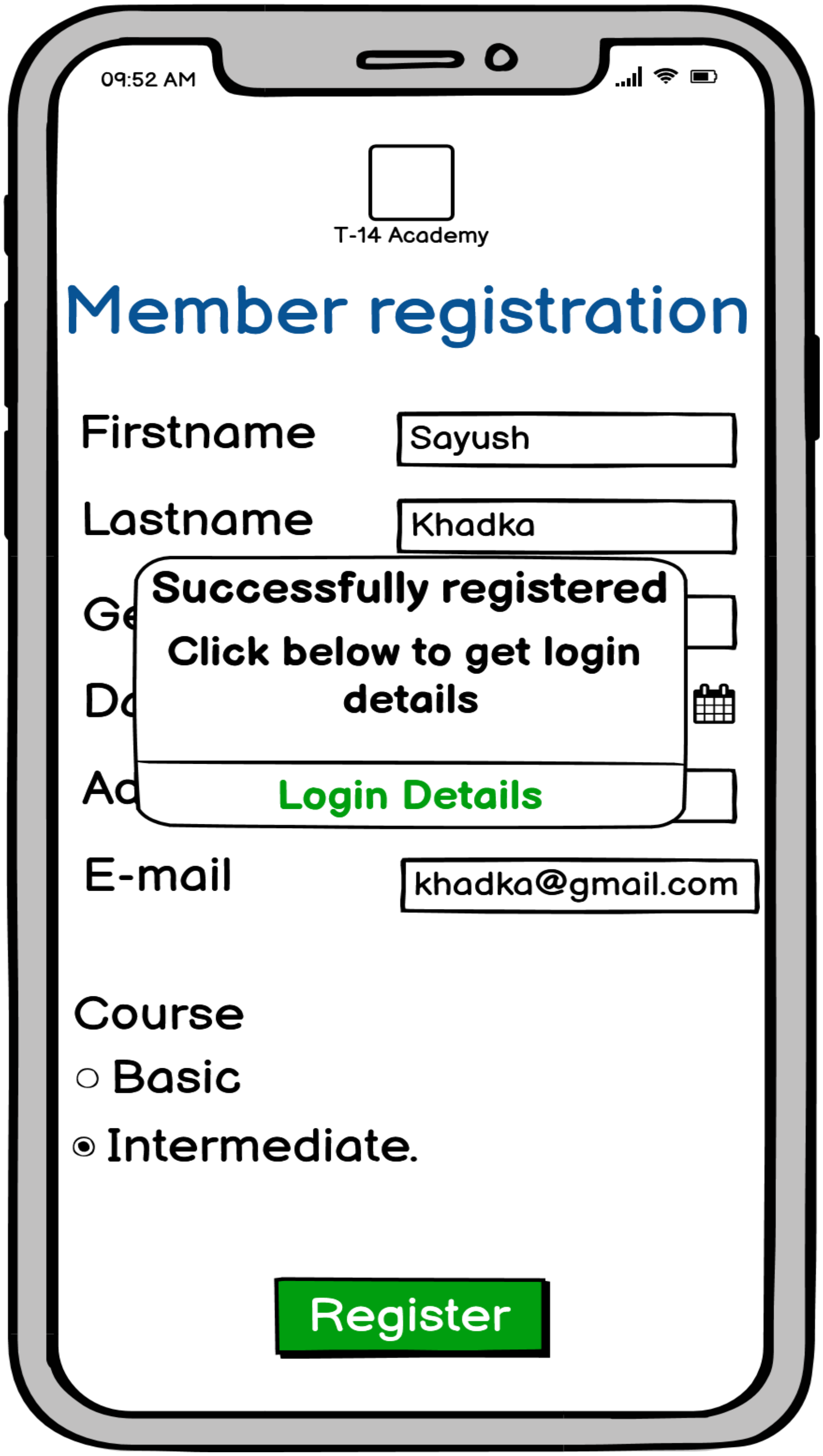
Course

○ Basic

● Intermediate.

Figure 11 Prototype: Register membership (Valid registration information)

7.7 Prototype: Register membership (Successfully registered)



Firstname Sayush

Lastname Khadka

Ge

Do

Ad

E-mail khadka@gmail.com

Course

○ Basic

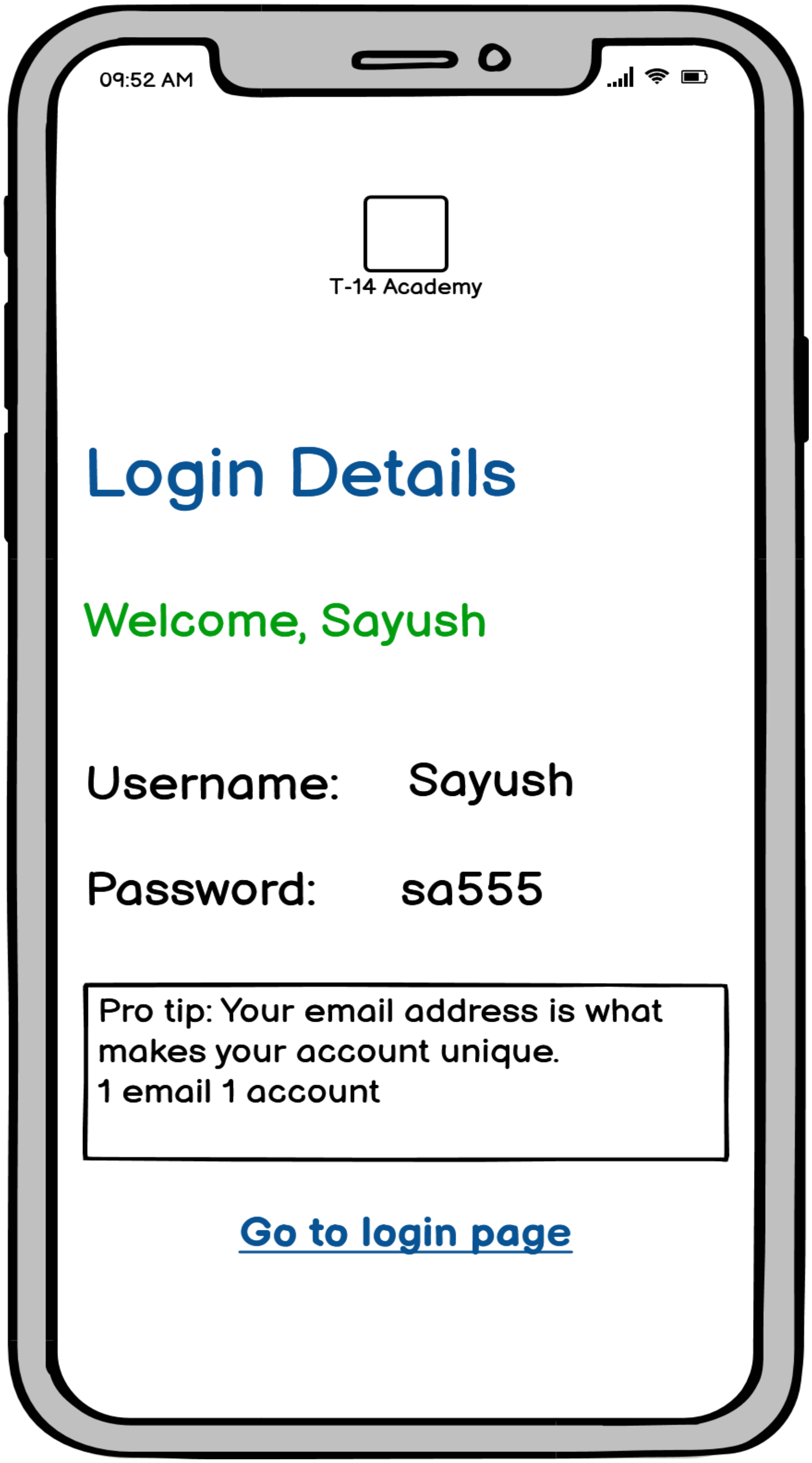
● Intermediate.

Successfully registered

Click below to get login details

Figure 12 Prototype: Register membership (Successfully registered)

7.8 Prototype: Login details provided



Login Details

Welcome, Sayush

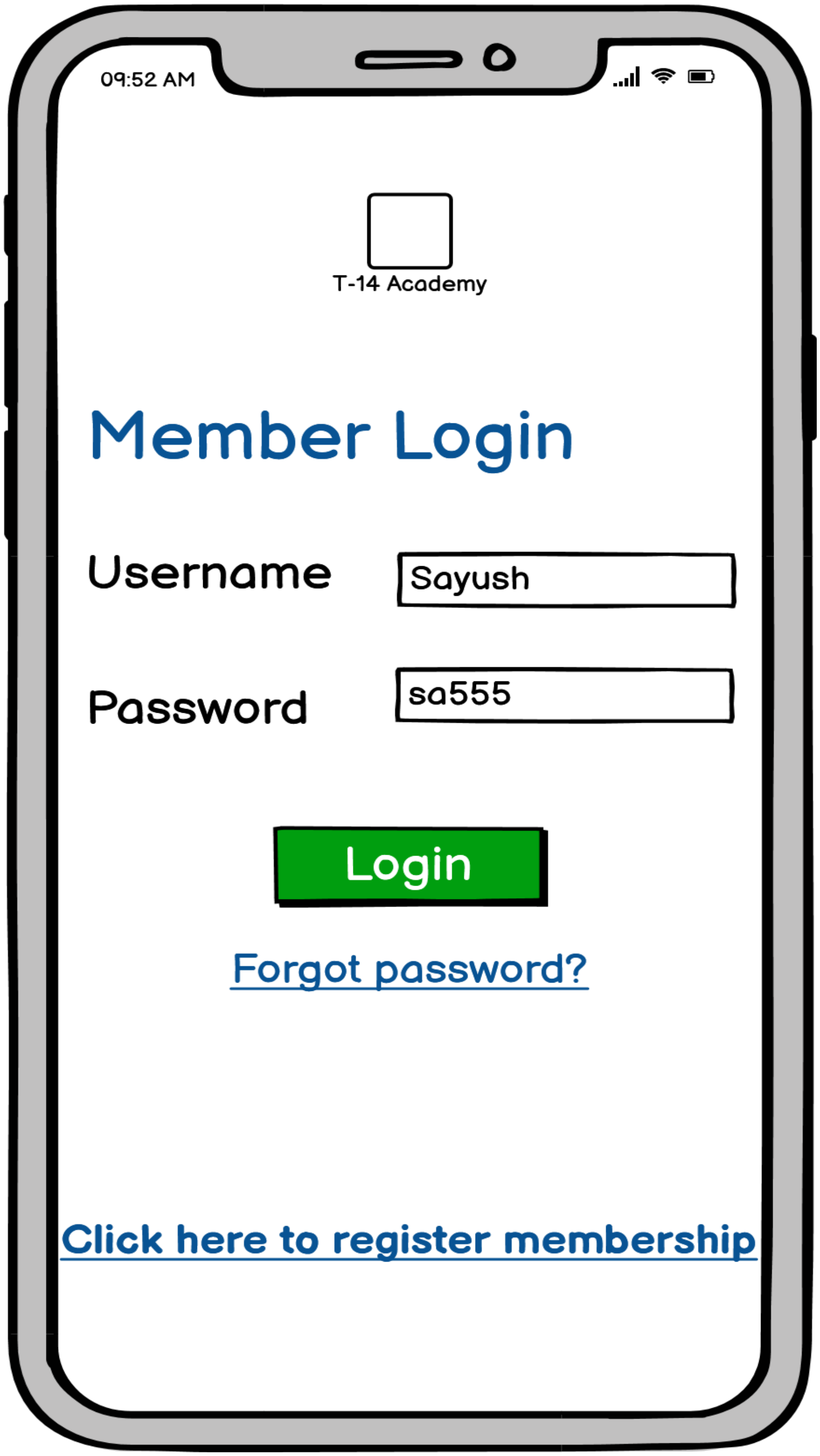
Username: Sayush

Password: sa555

Pro tip: Your email address is what makes your account unique.
1 email 1 account

Figure 13 Prototype: Login details provided

7.9 Prototype: Member login (Valid login information)

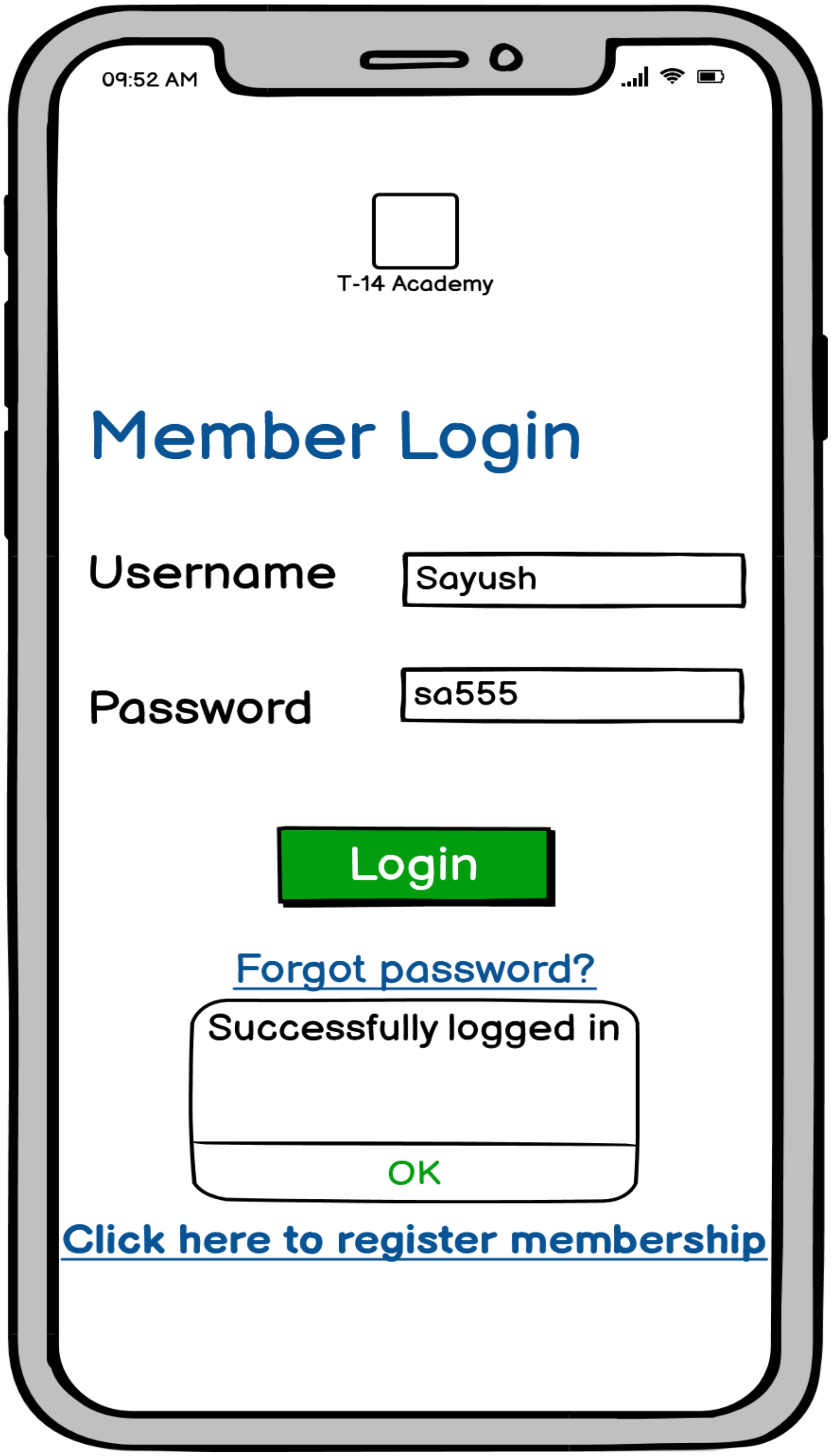


Username: Sayush

Password: sa555

Figure 14 Prototype: Member login (Valid login information)

7.10 Prototype: Member login (Login successful)



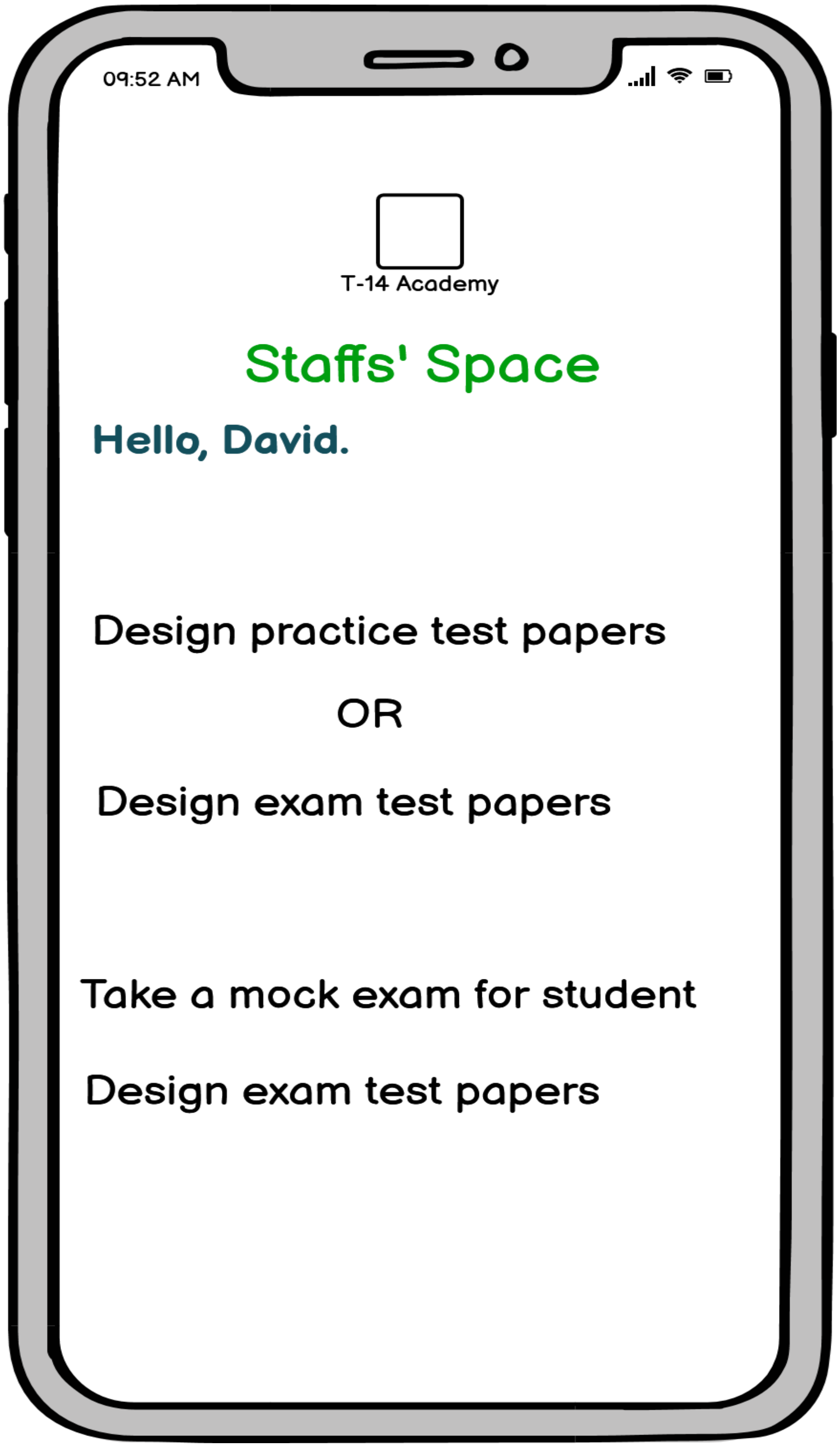
Username Sayush

Password sa555

Successfully logged in

Figure 15 Prototype: Member login (Login successful)

7.11 Prototype: Staffs' Space (Design exam test papers and practice test papers)



09:52 AM

T-14 Academy

Staffs' Space

Hello, David.

Design practice test papers

OR

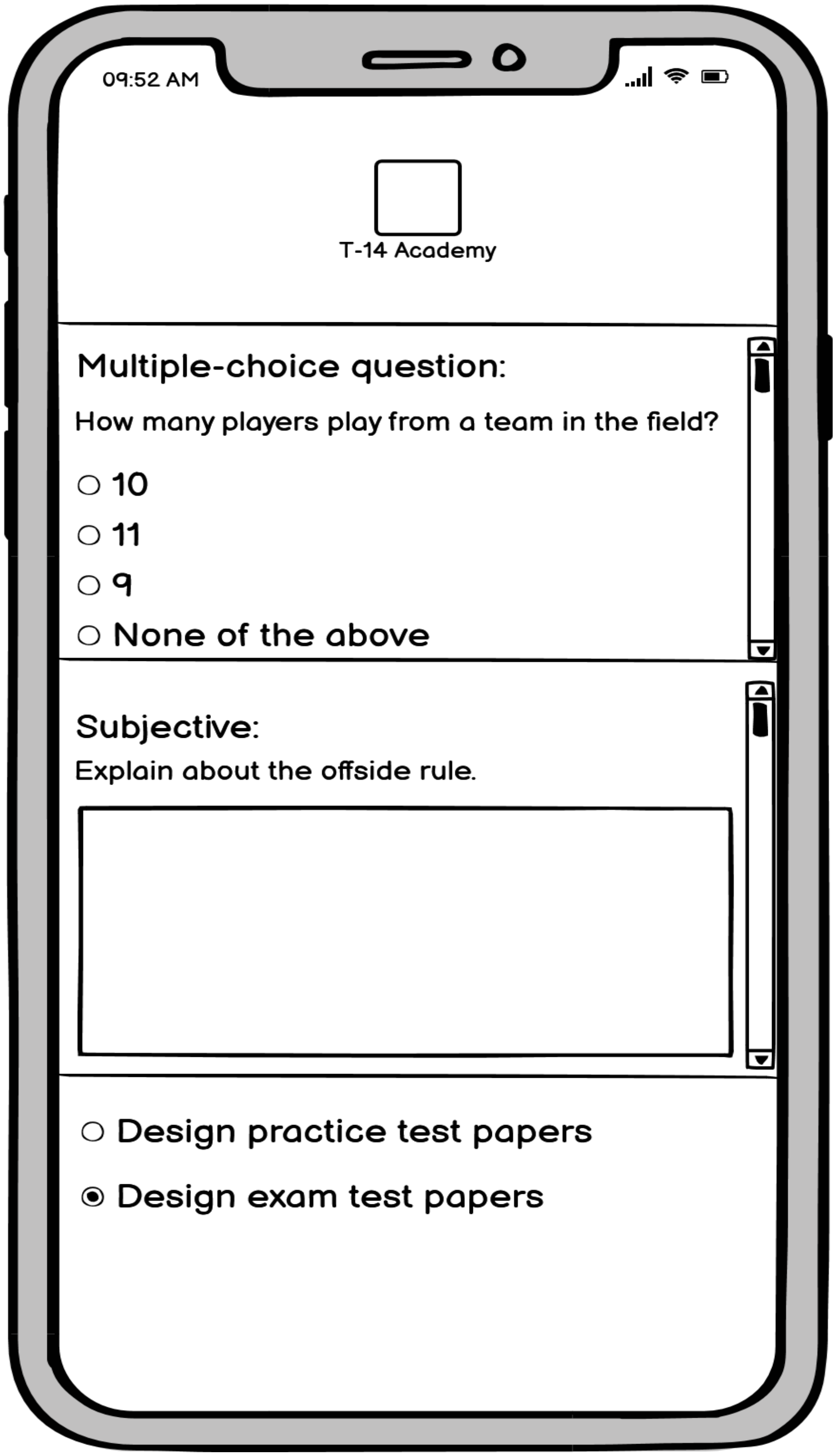
Design exam test papers

Take a mock exam for student

Design exam test papers

Figure 16 Prototype: Staffs' Space (Design exam test papers and practice test papers)

7.12 Prototype: Staffs' Space (The provision to design test papers (MCQ, Subjective) in the system itself



Multiple-choice question:

How many players play from a team in the field?

○ 10
○ 11
○ 9
○ None of the above

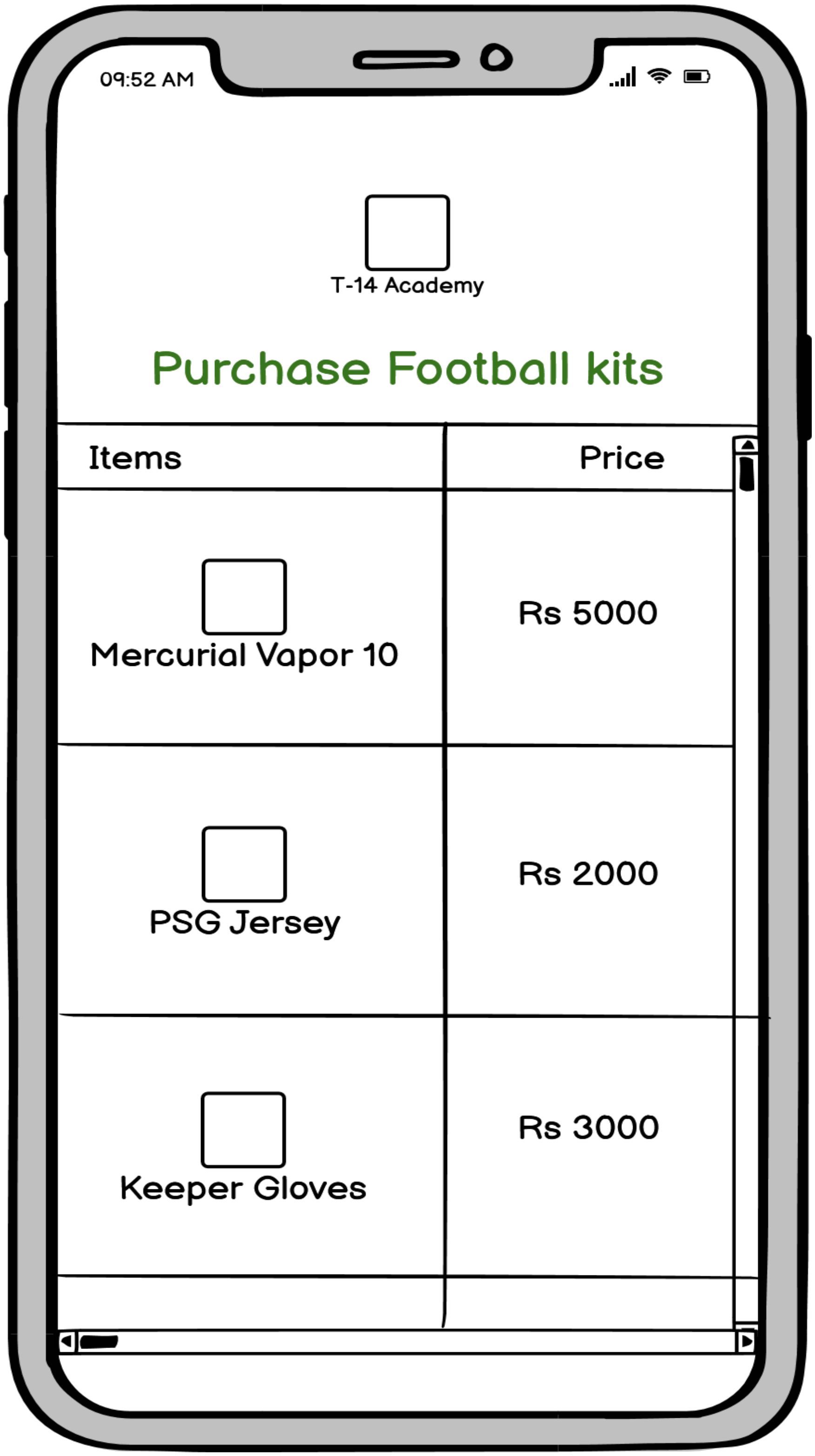
Subjective:

Explain about the offside rule.

○ Design practice test papers
● Design exam test papers

Figure 17 Prototype: Staffs' Space (The provision to design test papers (MCQ, Subjective) in the system itself

7.13 Prototype: Purchase football kits.

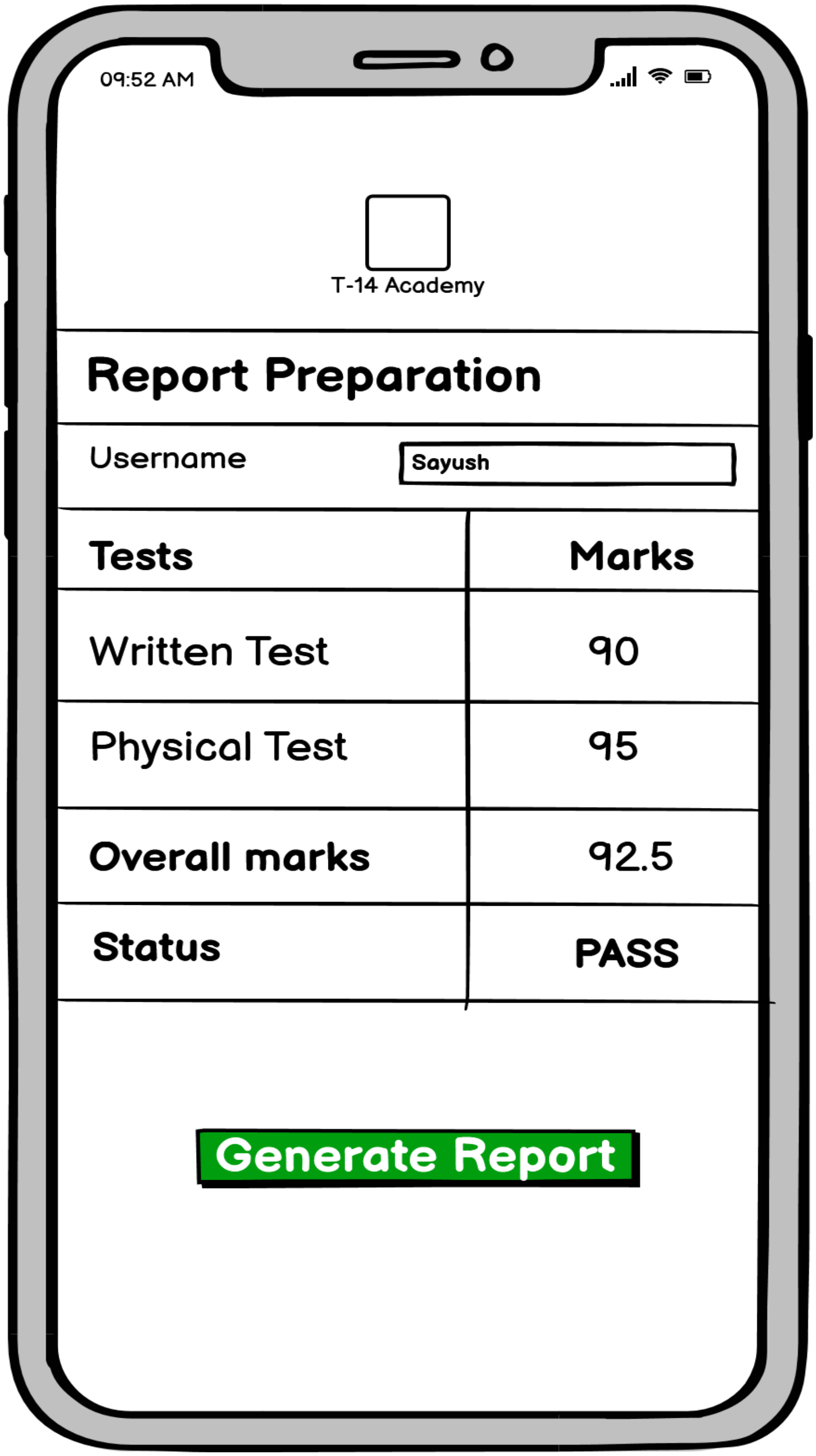


Purchase Football kits

Items	Price
Mercurial Vapor 10	Rs 5000
PSG Jersey	Rs 2000
Keeper Gloves	Rs 3000

Figure 18 Prototype: Purchase football kits

7.14 Prototype: Report preparation and generation



Report Preparation

Username: Sayush

Tests	Marks
Written Test	90
Physical Test	95
Overall marks	92.5
Status	PASS

Figure 19 Report preparation and generation

7.15 Prototype: Admin's Space (Post exams notices and announcements)

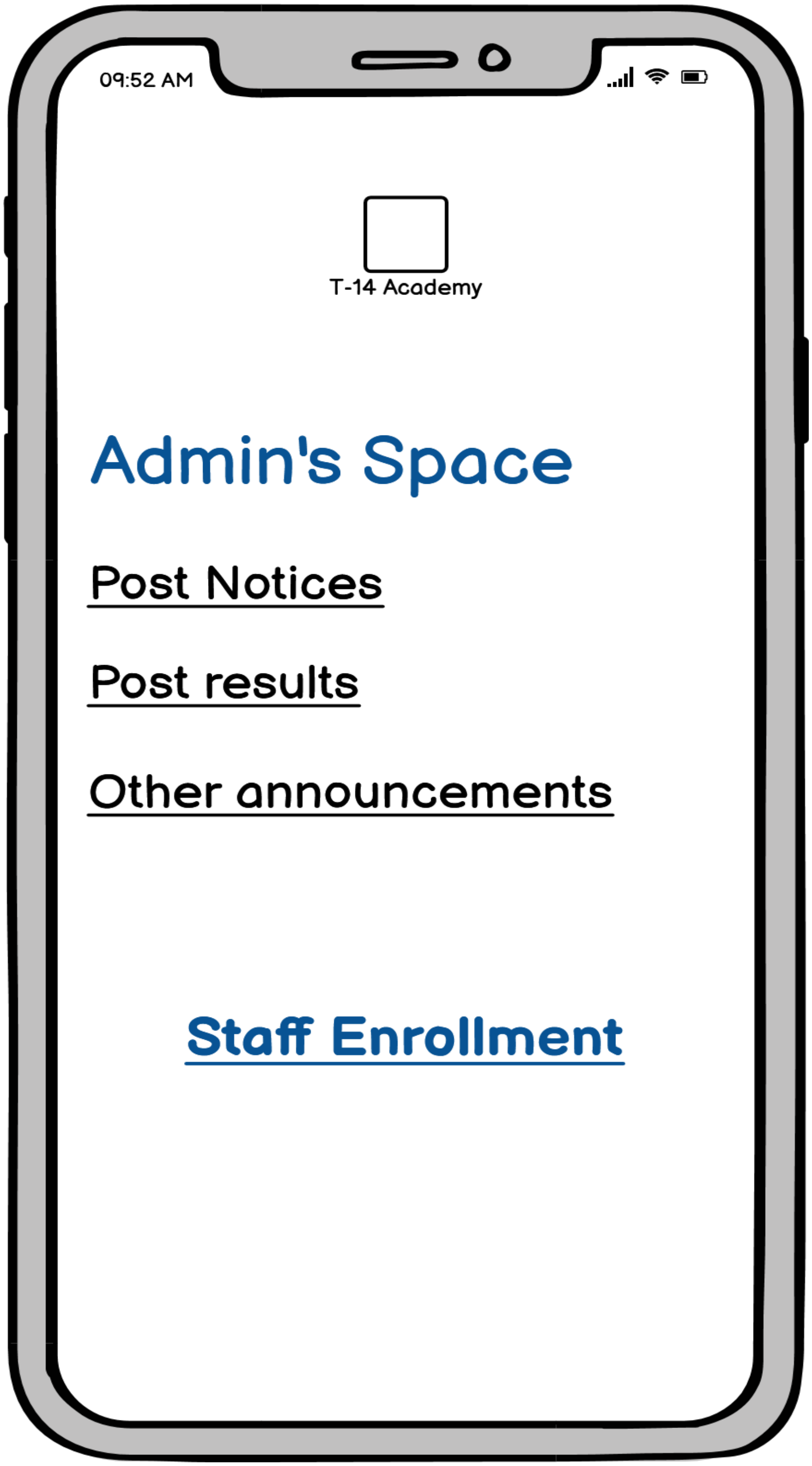
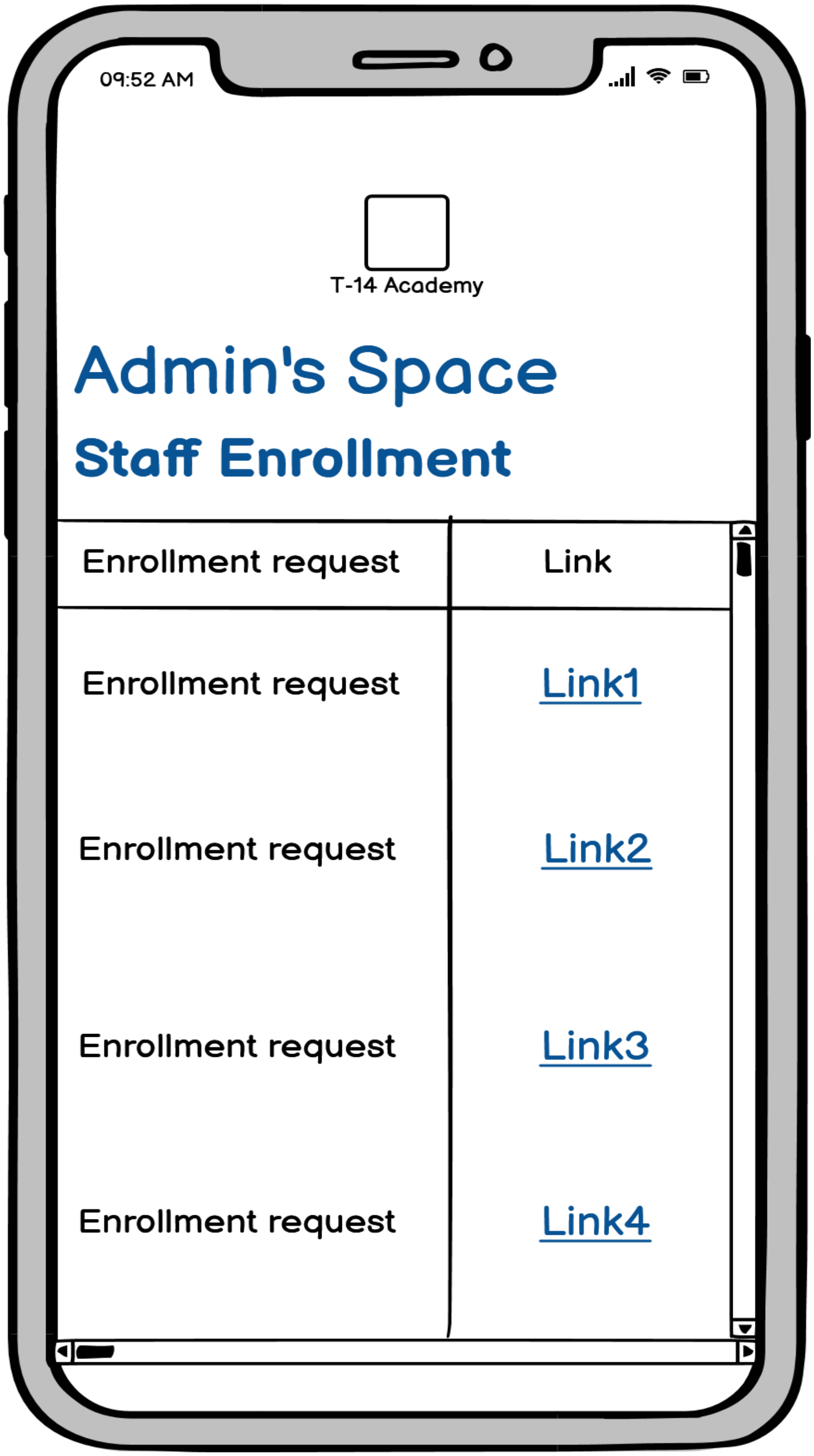


Figure 20 Admin's Space (Post exams notices and announcements)

7.16 Prototype: Enroll staff members



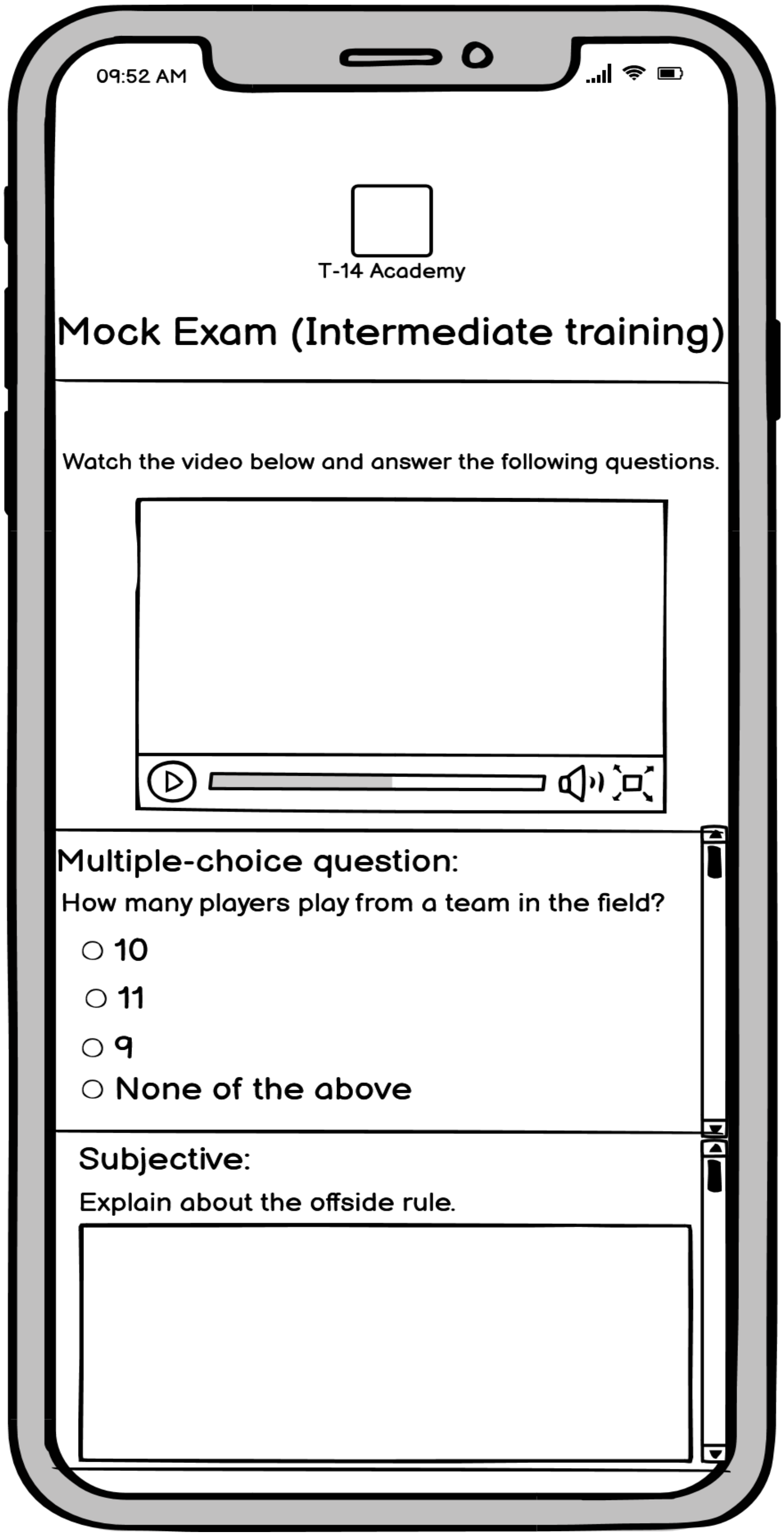
Admin's Space

Staff Enrollment

Enrollment request	Link
Enrollment request	Link1
Enrollment request	Link2
Enrollment request	Link3
Enrollment request	Link4

Figure 21 Admin's Space (Enroll staff members)

7.17 Prototype: Take a mock exam



09:52 AM
T-14 Academy
Mock Exam (Intermediate training)
Watch the video below and answer the following questions.
Multiple-choice question:
How many players play from a team in the field?
○ 10
○ 11
○ 9
○ None of the above
Subjective:
Explain about the offside rule.

Figure 22 Prototype: Take a mock exam

8. Conclusion:

The coursework was a great success with the acquisition of knowledge on Object-Oriented Analysis and Design. The project was fruitful and instructive, showing us how to create different diagrams related to the concept of OOA and design straightforwardly and effectively. The project was developed in response to a real-world difficulty, and it may be used in designing and developing the system for different sectors.

The product was made even more interesting by designing the prototype (wireframe) for the system. It helped us realize and visualize the real implementation of the design and development and the overall outlook of the design process of the system made for T -14 Academy. This coursework had some similarities with the first one in the sense that the given scenario was the same but differed in the implementation of the project. The project assisted us in the process of specialization of the important factors in a system development process like making a Gantt chart, communication diagrams (Collaboration diagram and Sequence diagram), and class diagram. Implementing OOA and selecting a methodology feasible for the system was also carried out. The Concept of architecture pattern and testing method of the system has also been specified, researched, and discussed.

Problems always occur when it comes to building a project. In similar terms, there are some difficulties while doing this project too. One of the difficult and time-consuming tasks was frequent changes or adjustments in the different diagrams according to the client’s assumption. The notion and the concepts were new and were quite perplexing in some cases. But these difficulties were overcome by detailed research work and implementation of the research done. These difficulties were helped to be overcome by the teachers and their guidance. Whenever needed, the teachers were there to help us. The project required a significant amount of time and work, but it allowed us to gain a better understanding of the concept of OOA (Object Oriented Analysis) and Design.

Bibliography

Alam, R., 2019. *Cybarlab.* [Online]
Available at: https://cybarlab.com/ooa
[Accessed 2 May 2022].

Contributor, T., 2020. *TechTarget.* [Online]
Available at: https://www.techtarget.com/whatis/definition/use-case-diagram
[Accessed 4 May 2022].

Dhaduk, H., 2020. *SIMFORM.* [Online]
Available at: https://www.simform.com/blog/software-architecture-patterns/
[Accessed 8 May 2022].

EdrawMax, W., n.d. *Wondershare EdrawMax.* [Online]
Available at: https://www.edrawmax.com/article/communication-diagram-uml.html
[Accessed 6 May 2022].

Grant, M., 2021. *Investopedia.* [Online]
Available at: https://www.investopedia.com/terms/g/gantt-chart.asp
[Accessed 3 May 2022].

Hamilton, T., 2022. *Guru99.* [Online]
Available at: https://www.guru99.com/white-box-testing.html
[Accessed 8 May 2022].

javaTpoint, n.d. *javaTpoint.* [Online]
Available at: https://www.javatpoint.com/use-case-model
[Accessed 3 May 2022].

javaTpoint, n.d. *javaTpoint.* [Online]
Available at: https://www.javatpoint.com/uml-collaboration-diagram
[Accessed 6 May 2022].

Lewis, S., 2019. *Tech Target.* [Online]
Available at: https://www.techtarget.com/searchcio/definition/Prototyping-Model
[Accessed 2 May 2022].

Martin, M., 2022. *Guru99.* [Online]
Available at: https://www.guru99.com/software-engineering-prototyping-model.html
[Accessed 8 May 2022].

smartdraw, n.d. *smartdraw.* [Online]
Available at: https://www.smartdraw.com/sequence-diagram/
[Accessed 6 May 2022].

techopedia, n.d. *techopedia.* [Online]
Available at: https://www.techopedia.com/definition/16466/class-diagram
[Accessed 8 May 2022].

Wayner, P., n.d. *TechBeacon.* [Online]
Available at: https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice
[Accessed 8 May 2022].