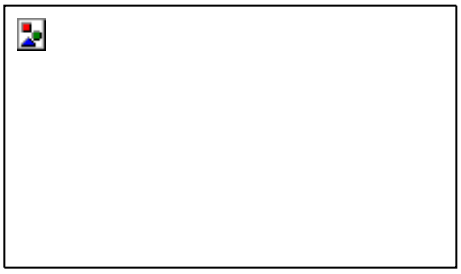


Module Code & Module Title

CS5002NI SOFTWARE ENGINEERING

Assessment Weightage & Type

35% Individual Coursework

Year and Semester

2021-22 Spring

Student Name:Trishala Prasai

London Met ID:20049085

College ID: NP01CP4S210317

Assignment Due Date: May 9 2022

Assignment Submission Date:May 9 2022

Title (Where Required): T-14 Academy

Word Count (Where Required):3393

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.

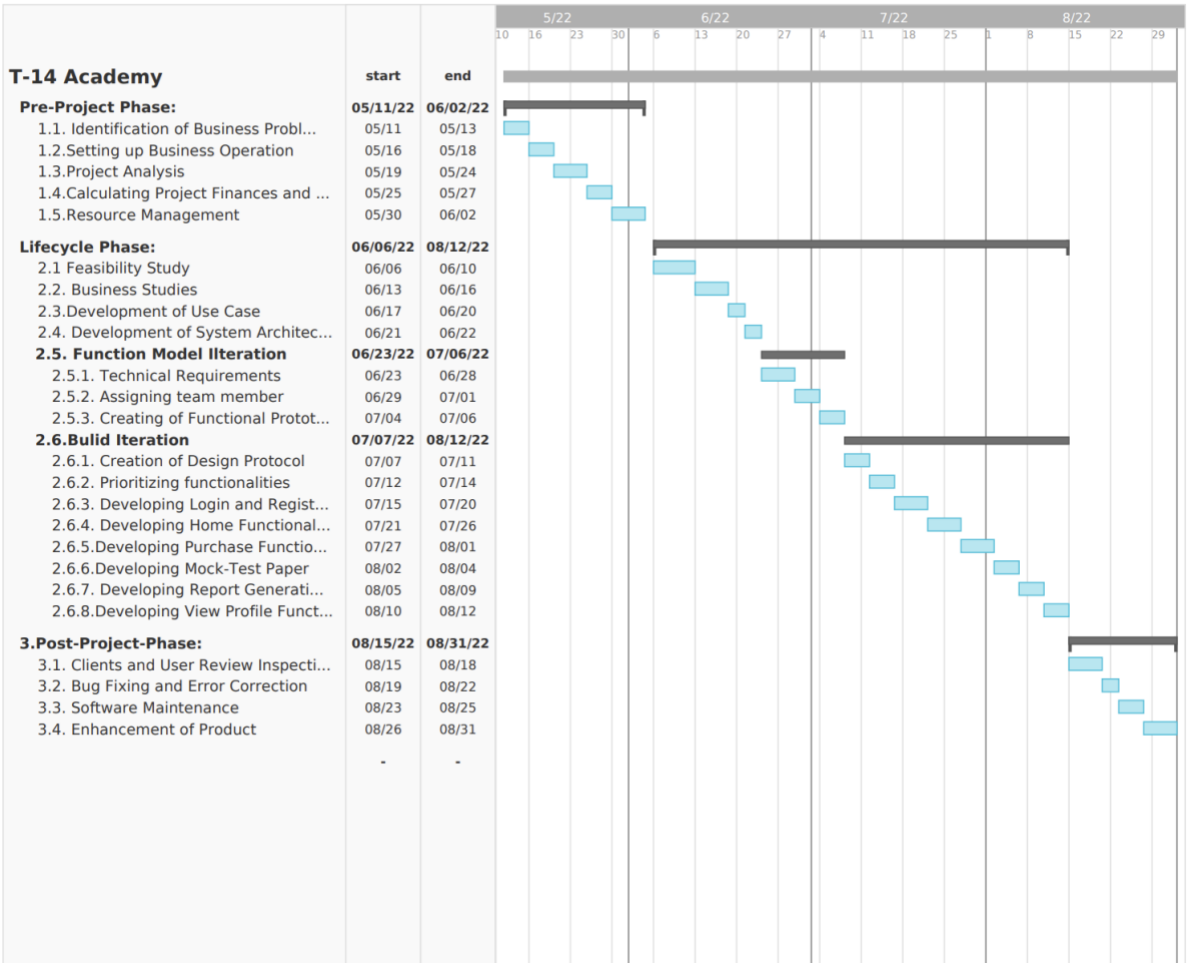
Table of Contents

Introduction

This coursework is the second coursework done, with this coursework being valued at 35% of the total grades. In this coursework I have developed a online based computerized information system of the Academy “T-14 Academy” which is a academy for football training and sells Football Accessories at a price lower than that of the market.

2.Gantt Chart

Timelines and tasks from project management are translated into a horizontal bar chart that shows start and finish dates, along with dependencies, scheduling, and deadlines, as well as how much of the job is accomplished each stage and who is the task owner. When there is a big team and various stakeholders involved, this might help keep work on schedule. A Gantt Chart makes it simple to represent Tasks, Subtasks, Milestones, and Projects on a graph. It is commonly used in heavy industries for projects such as dam construction, bridge construction, and highway construction, along with software development and the creation of other goods and services. (APM, 2022)



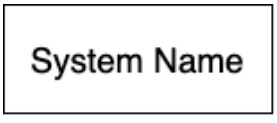
Conducting research, picking a software product, testing the program, and installing it are all project activities that must be completed if the project involves installing new software on a server. The software selection is a significant milestone. On the graph, these tasks show as vertical lines. (Analysitabs, 2022)

3.Use Case Model

The use-case model is a diagram that depicts how people interact with a system to solve a problem. As a result, the use case model specifies the user's goal, the system's interactions with the user, and the system's behavior necessary to achieve these goals. The various model elements are actor, use cases, and the association between them. A use-case model, which is used to arrange the model to facilitate analysis, planning, navigation, communication, development, and maintenance, may be included in a package. The use-case model serves as a integrated thread throughout the system's development.

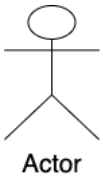
The use-case model serves as the core specification of the system's functional needs, as well as a foundation for design and analysis, user documentation, test case definition, and iteration planning.

System: A rectangle box represents the system. Everything in the system, with the exception of actors, functions perfectly in rectangular box.



Use Case: The function of the system is represented by the use, which is symbolized by the oval form.

Actor: Users of the system created are actors.



Relations: The arrow with a label in it is used to represent all of the relationships.

Use-case diagrams represent the behavior of a system and aid in the capturing of its needs. Use-case diagrams depict a system's high-level functionality and scope. The interactions between the system and its actors are also depicted in these diagrams. In use-case diagrams, the use cases and actors define what the system does and how the actors interact with it, but not how the system works inside. A single use-case diagram can be used to represent a complicated system, or numerous use-case diagrams can be used to model the system's components. Use-case diagrams are often created in the early stages of a project and referred to throughout the development process.


Validate Username and Password
<<includes>>
Display Error Message
<<extends>>
Log-In
If username or password is incorrect,Error message is displayed
Enroll Members
Admin
Register
Register by Gmail
Register by Contacts
Posts Notice
Result
<<includes>>
If valid registration details is not given,Error message is displayed
<<extends>>
Validate Registration Details
Clients
Display Error Message
MCQ
Design Questions
Objective
Purchase
Check sufficient balance
<<includes>>
Payment Method
Display error message
<<extendes>>
Staff
Registered Members
New Users
Give Mock Exam
If their is no sufficient balance,Error message is displayed
Physical
Report Prepration
Online
View Profile
Payment from e-sewa
Payment from pay-pal

3.1. High level use case description

3.1.1. Use case: Log-In

Actor: Clients

Description: The clients get access in the T-14 Training Academy by using valid Username and Password which is provided while registration.

3.1.2. Use case: Register

Actor: Clients

Description: The clients gets registered in the T-14 Training Academy by providing all the information required for the registration paying the required amount.

3.1.3. Use case: Give Mock Exam

Actor: Clients

Description: The clients is allowed to give pre-test before the examination.

3.1.4. Use case: View Profile

Actor: Client

Description: The clients get registered only after payment. The clients get to view their profile.

3.1.5. Use case: Purchase

Actor: Client

Description: The clients get registered only after payment. The clients pays amount from two different methods (E-sewa and Pay-Pal),if they have sufficient amount of money or else error message will be displayed.

3.1.6. Use case: Enrollment Member

Actor: Admin

Description: The admin can enroll members in the academy after collecting all the required information

3.1.7. Use case: Posts Notice

Actor: Admin

Description: Notices of the academy is posted by the admin.

3.1.8. Use case: Design Questions

Actor: Staff

Description: All the question for the test is prepared by the Staffs.

3.1.9. Use case: Report Preparation

Actor: Staff

Description: Report after the test is also prepared by the staff which can be physical or online.

3.2. Expanded use case description

Use case: Register

Actor: Clients

Description: All the required information are provided while registering in the T-14 Training Academy.

Typical courses of Events:

Actor Action	System Response
1.The clients Log-In UI from the Log-In page of the system.	
	2.The system navigates the clients to the systems Home Page.
3.The user clicks to the registration form button.	
	4.The system navigates the user to the registration form page.
5.The user fills the necessary details which is needed for the registration from i.e, User Name, Gmail, DOB, Password.	
	6.Verifies all the information of clients and proceeds to the verification process.
	7.The system takes the user to the payment method.

	8.The system provides many different ways of payment methods for payment
9. The user selects the payment method and necessary bank details and method of payment	
	10.The system verifies the payment and confirms registration.

Alternative:

Line 5: If the clients personal details is not given correctly the system displays error message and asks again to enter the valid details and the use case ends.

Line 9: If the clients cash is not sufficient ,the system displays error message about insufficient balance and the use case ends.

Use case: Purchase

Actor: Clients

Description: All the required information are provided while purchasing in the T-14 Training Academy.

Typical courses of Events:

Clients Action	System Response
1. The clients click to the purchase button.	
	2. The system navigates the clients to the system's purchase page.
	3.The systems shows the items in the purchase home page.
4.The user purchases some items.	
	5. The system requests it for payment method.
6.The user chooses the payment method of for the item and fill the required information, ie. Account Number, Pin Code.	
	7.The payment will be verified after choosing the method and giving the necessary details
	8.The system will display Successful Payment message.

Alternative:

Line 8: If the client’s cash is not sufficient, the system displays error message about insufficient balance and the use case ends.

4. Collaboration Diagram

In the Unified Modeling Language, a collaboration diagram, also known as a communication diagram, depicts the links and interactions among software components (UML). These diagrams can be used to depict the dynamic behavior of a certain use case and define each object's purpose.

The structural pieces required to carry out the functionality of an interaction are first identified before creating a collaboration diagram. The relationships between those parts are then used to create a model. Software for developing and editing collaborative diagrams is available from several providers.

(Tech Target, 2022)

Domain Class

Use case	Domain Class
Purchase Kits	User, Football Kits,Payment

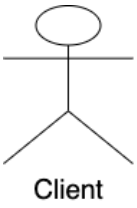
Controller Object (Same as Use case)

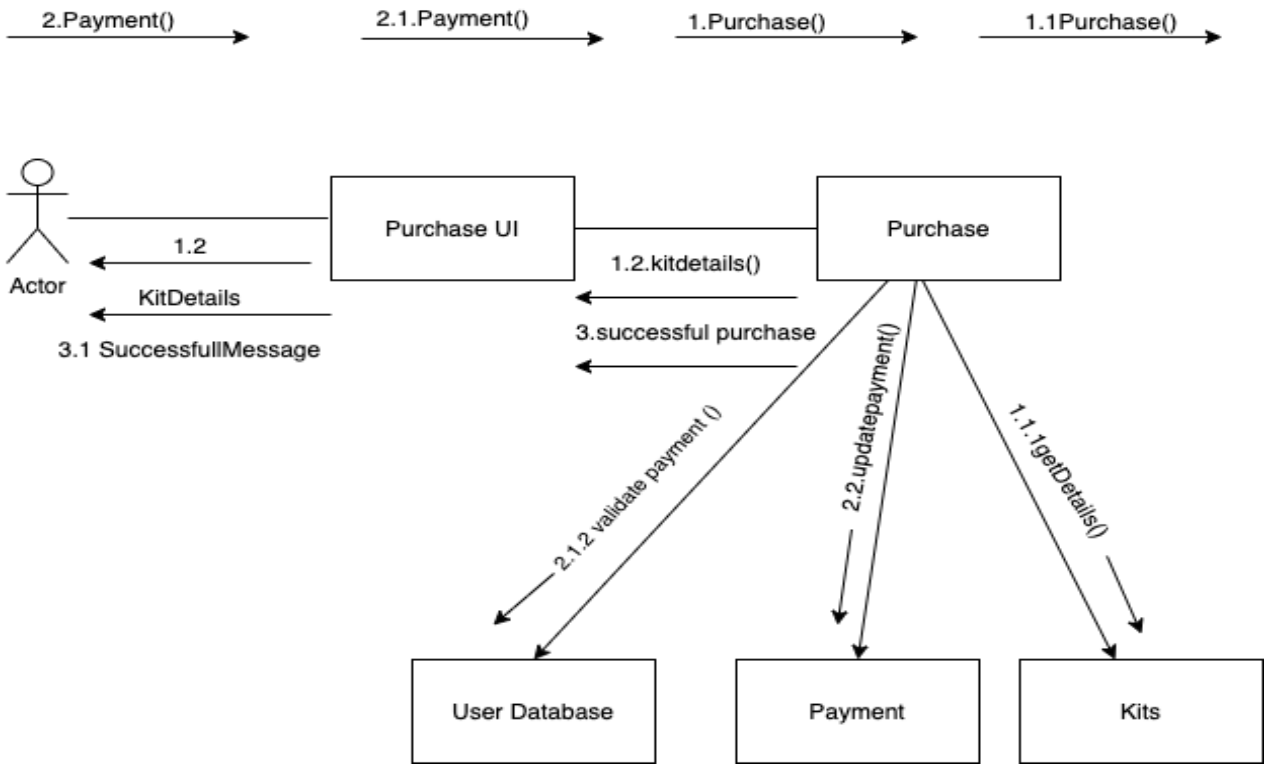
:PurchaseKit

Boundary object

:PurchaseKitUI

Actor



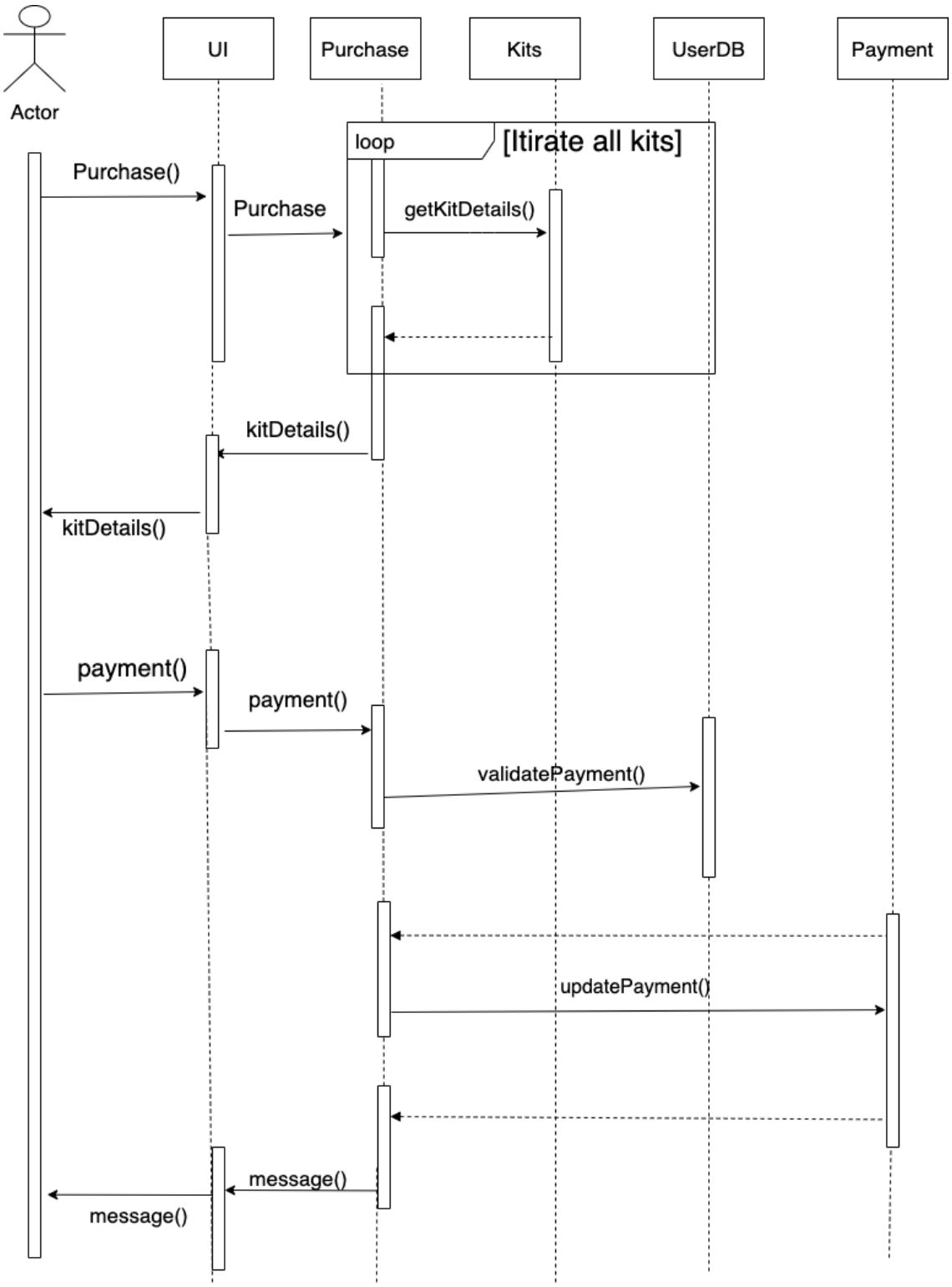


\

5.Sequence Diagram

To comprehend what a sequence diagram is, you must first comprehend the purpose of the Unified Modeling Language, or UML. UML is a modeling toolset that may be used to create and notate a variety of diagrams, such as behavior diagrams, interaction diagrams, and structural diagrams.

A sequence diagram is a form of interaction diagram that shows how a group of items interacts and in what order. Software engineers and business experts use these diagrams to understand the requirements for a new system or to describe an existing process. Event diagrams and event scenarios are other names for sequence diagrams. UML Sequence Diagrams show how operations are carried out. They depict object-to-object interaction in a collaborative setting. Sequence Diagrams are time focused and depict the interaction order visually by using the vertical axis of the diagram to represent time.


Actor
UI
Purchase
Kits
UserDB
Payment
loop
[Itirate all kits]
Purchase()
Purchase
getKitDetails()
kitDetails()
kitDetails()
payment()
payment()
validatePayment()
updatePayment()
message()
message()

6.Class Diagram

The class diagram depicts the static perspective of an application. This class diagram depicts various sorts of objects in the system as well as their relationships. Different components of the digaram class include:

Dependency:

A semantic line connecting two or more classes, where modifications in one class cause changes in the others.

Generalization:

A semantic line connecting two or more classes, where modifications in one class cause changes in the others.

Association:

A connection that is either static or physical between two or more items. It shows how many objects are involved in the relationship.

Fi

A static diagram is a class diagram. It depicts an application's static view. A class diagram is used not only for visualizing, describing, and documenting many parts of a system, but also for creating executable code for a software program.

A class diagram depicts a class's attributes and operations, as well as the system's limitations. Because class diagrams are the only UML diagrams that can be directly mapped with object-oriented languages, they are frequently utilized in the modeling of object-oriented systems.

A collection of classes, interfaces, affiliations, collaborations, and constraints are shown in a class diagram. It is also known as a structural diagram.

7. Further Development

The main purpose of this coursework is to develop a system for a T-14 Training Academy. It is a system made for football training. The academy offers a variety of training programs for people of all ages. There are two sorts of training categories available for each age group: Basic and Intermediate. Basic training does not have any precise requirements. However, there are some qualifications that must be met before enrolling in the intermediate training. To be considered for intermediate training, one must take some football IQ exams and score certain points to be considered for other tests. After analyzing the total performance in both written and physical tests, the trainer team determines whether or not the members are eligible for Intermediate training. In addition to the training facilities, it also provides Football Accessories at a lower price than the market.

The system we are about to make is object oriented in which we used The Dynamic Systems Development Method (DSDM) is an agile project delivery paradigm that was first developed in 1994 for software development. It was supposed to be a step forward from Rapid Application Development (RAD), which focused on rapid prototyping and iteration based on user feedback. The DSDM Agile Project Framework, like many other agile project delivery methodologies, evolved from a software-specific solution to a more general project management tool.

The Dynamic Systems Development Method includes the following elements:

The dependence on robust foundations and governance sets it different from other methods.

Progress is made incrementally and iteratively.

Feedback from users or customers is essential for continuous improvement.

Cost, quality, and time limits are all important considerations.

Organizes scope into Must Have, Should Have, Could Have, and Won't Have categories.

The DSDM agile principles are the guiding force behind every project. There are 8 principles in total.

Focus on the business need

Deliver on time

Collaborate

Never compromise quality

Build incrementally from firm foundations

Develop iteratively

Communicate continuously and clearly

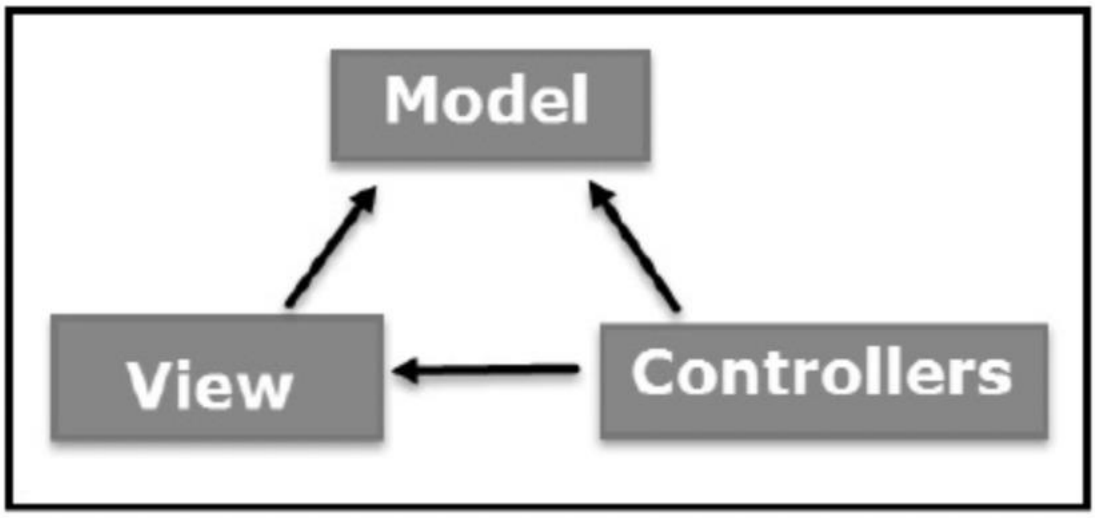
Demonstrate control

(The qalead, 2022)

Selecting an architecture design:

The coding process is one of the most difficult parts for a developer after the design process. For choosing the coding part MVC is an architectural paradigm that divides an application into three logical components: model, view, and controller. Each of these components is designed to handle certain parts of application development. MVC is a popular industry-standard web development framework for developing scalable and flexible projects.

Components of MVC



Model

All data-related logic that the user engages with is represented by the Model component. This can be the data being transmitted between the View and Controller components or any other data related to business logic. A Customer object, for example, will retrieve

customer information from a database, change it, and either update or output the data back to the database.

View

All of the application's UI logic is handled by the View component. The Customer view, for example, will have all UI components that the final user interacts with, such as text boxes, dropdowns, and so on.

Controllers

Controllers serve as a link between the Model and View components, processing all business logic and incoming requests, manipulating data using the Model, and interacting with Views to produce the final output. The Customer controller, for example, will handle all interactions and inputs from the Customer View and use the Customer Model to update the database. The Customer data will be seen using the same controller. (Tutorials point, 2022)

Testing in development phase:

Effective and productive testing necessitates collaboration from all project stakeholders to improve the efficiency of the test-fix-retest cycle. This idea is in line with the DSDM collaboration approach, and it should include people from the business and technical, solution development, and testing.

Testing is usually a continuous aspect of the project throughout the SDLC. As a result, regardless of the phase you are now in, it is always involved in SDLC in some form.

The testing methods' main purpose is to report, monitor, troubleshoot, and retest software components until they meet the quality requirements established in the initial SRS. Many teams use papers like RTM to track the project's primary needs and analyze how they interact.

Types of Testing in SDLC

The testing steps of the Software Development Life Cycle are among the most crucial. To ensure that all requirements are met, these processes must be carried out in a thorough manner. System testing, integration testing, acceptance testing, and unit testing are the four basic phases of software testing, according to software testers.

Unit Testing

Unit testing is done on smaller software components that testers refer to as a single unit. Individual functions, code components, and even classes can be as little as these units, or as complex as entire software features. The testers can usually compile, load, and execute the smallest testable component of the software. Unit testing verifies that each component of the software is working properly.

Integration Testing

In integration testing, testers join multiple software modules and test them all at the same time. This type of testing guarantees that the system as a whole follows the correct data flow. These tests are required to ensure that the integrated system is ready to be tested.

System Testing

System testing is the process of combining multiple systems in integrated testing into a single integrated system. Here, testers assess the project's functional requirements once more, as well as whether the system adheres to the specified criteria. In addition, testers can assess how various components interact with one another. As a result, they can undertake particular testing processes on the integrated system, such as performance, load, reliability, and security testing.

User Acceptance Testing

Customers utilize software components to verify if it satisfies their criteria during user acceptance testing. If necessary, they might request that the development team make more improvements to the product. This final interaction with end-users and stakeholders can ensure that the product meets all of the agreed-upon requirements.

(performance syllabus, 2022)

Black Box testing

Black Box Testing is a software testing method that involves testing the functions of software applications without knowing the internal code structure, implementation details, or internal routes. Black Box Testing is a type of software testing that focuses on the input and output of software applications and is completely based on software requirements and specifications. Behavioral testing is another name for it.

Maintenance:

The process of upgrading, modifying, and updating software to stay up with client needs is known as software maintenance. After a product has been released, software maintenance is performed for a variety of purposes, including improving the software overall, addressing faults or bugs, increasing performance, and more.The SDLC includes software maintenance by default (software development life cycle). Software developers

don't have the luxury of releasing a product and leaving it alone; they must constantly correct and enhance their code in order to be competitive and relevant.

The four forms of software maintenance are carried out for various causes and objectives. Throughout its lifetime, a piece of software may require one, two, or all sorts of maintenance.

Corrective Software Maintenance

Corrective software maintenance is the most common and traditional type of maintenance (for software and anything else for that matter). When something goes wrong with a piece of software, such as bugs or errors, corrective software maintenance is required. These issues can have a wide influence on the software's overall functionality and must be corrected as soon as possible.

Due to bug reports sent in by consumers, software makers can often address issues that require corrective maintenance. If a company can detect and correct flaws before users notice them, it has a distinct advantage that makes it appear more credible and trustworthy (no one likes an error message after all).

Preventative Software Maintenance

Preventative software maintenance involves planning ahead to ensure that your program continues to function as intended for as long as possible.This includes making appropriate adjustments, enhancements, and modifications, among other things. Preventative software maintenance can address minor issues that may seem insignificant at the time but could grow into major concerns in the future. These are known as latent flaws, and they must be identified and repaired so that they do not become active faults.

Perfective Software Maintenance

Like any other product, when software is released to the public, new worries and ideas arise. Users may select additional software features or requirements to make the software the best tool for their needs. This is where perfect software maintenance comes in.

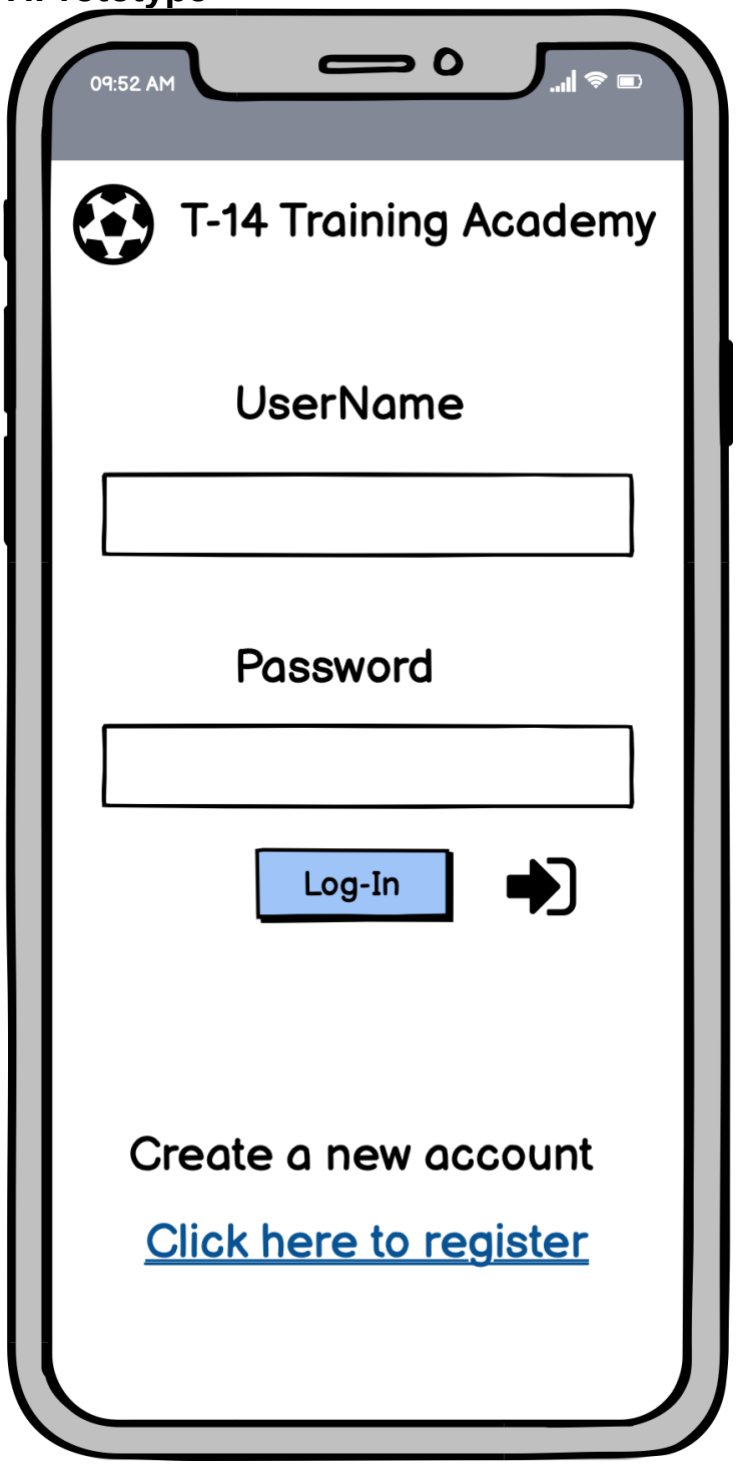
Perfective software maintenance aims to improve software by adding new features and removing components that are no longer needed. As the market and user needs change, so does software.

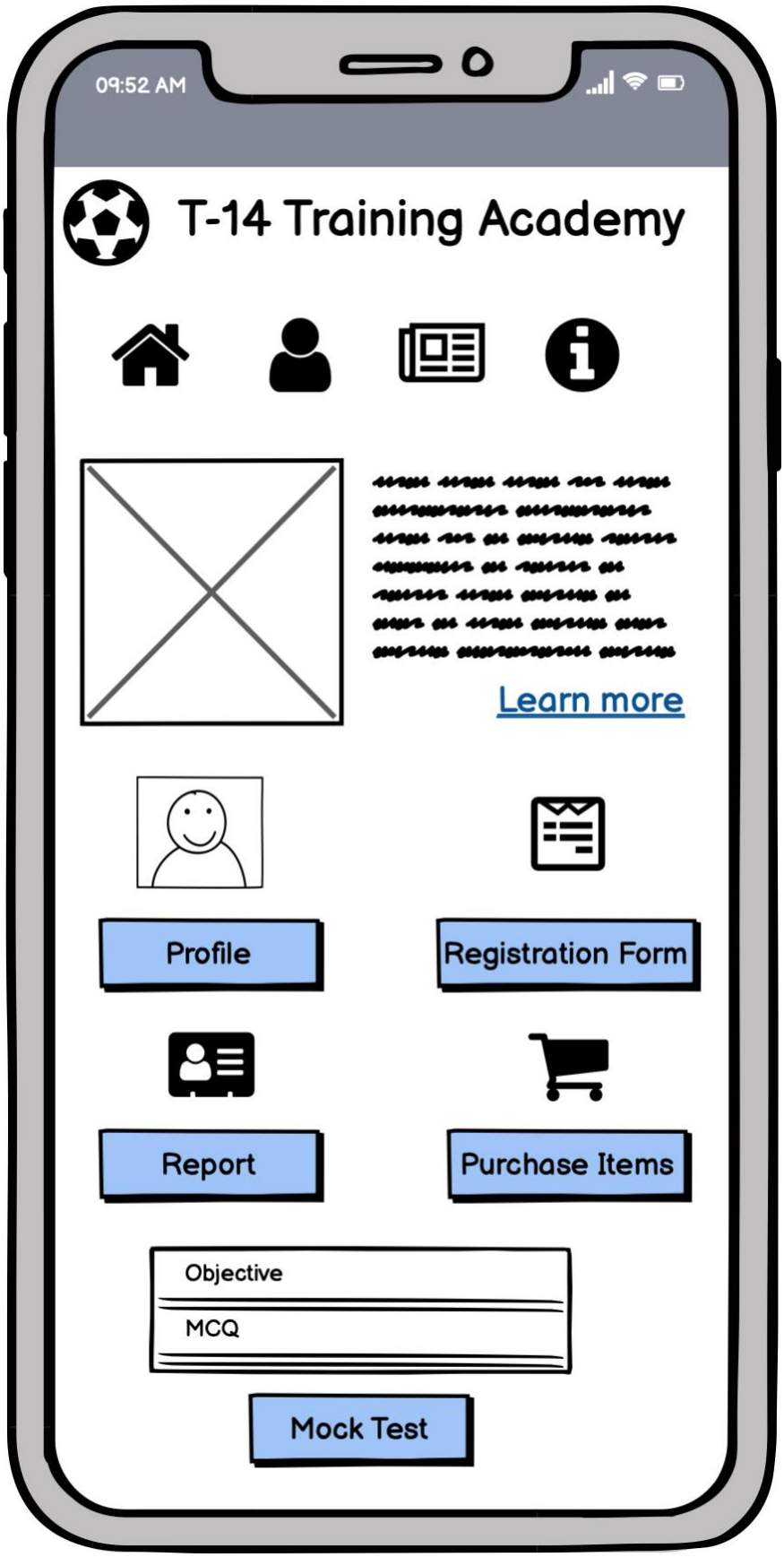
Adaptive Software Maintenance

Adaptive software maintenance is concerned with evolving technologies as well as software regulations and standards. These include updates to the operating system, cloud storage, and hardware. When these modifications are made, your program must adapt to match the new requirements and continue to function effectively.

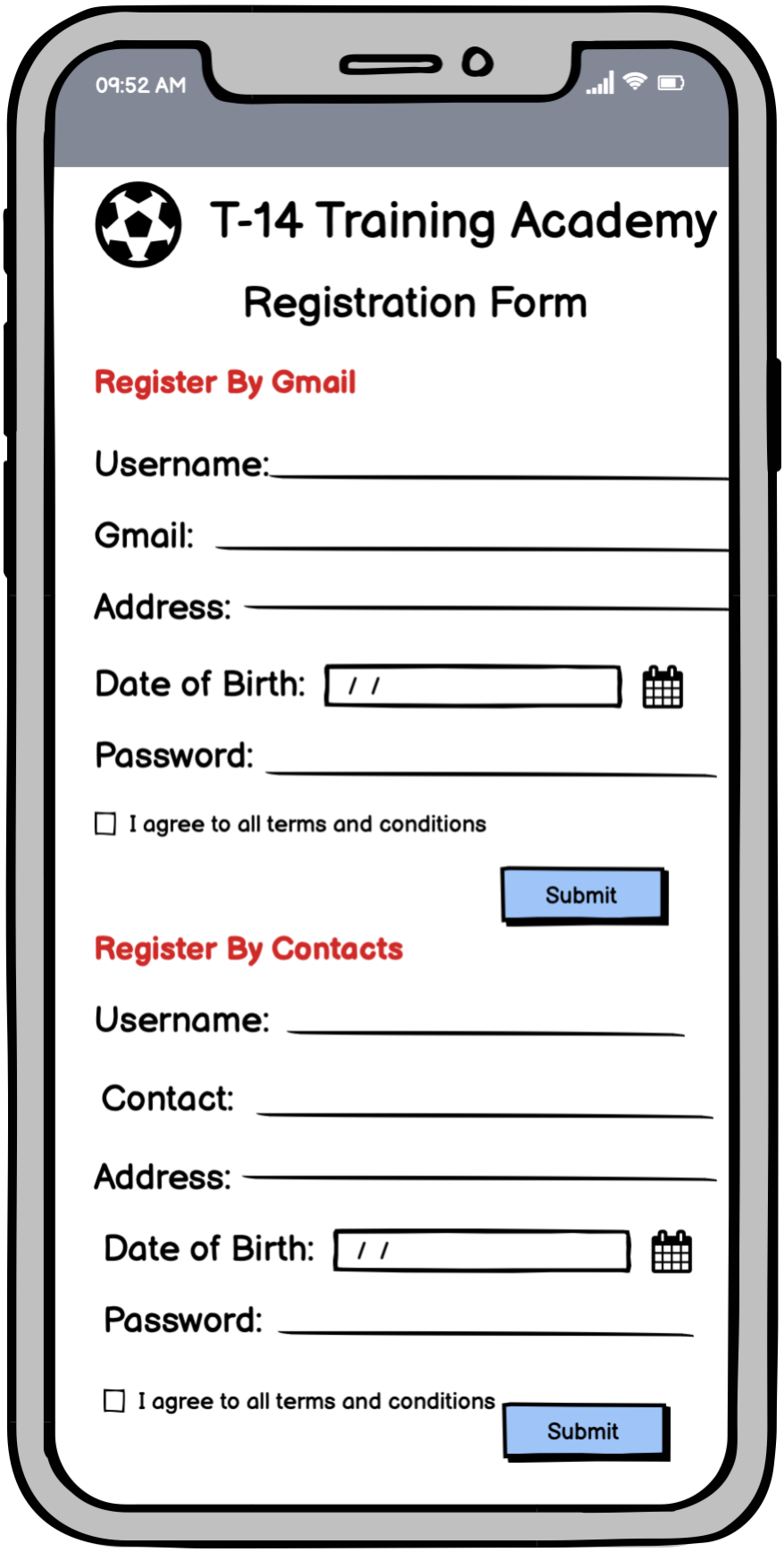
(Thalesgroup, 2022)

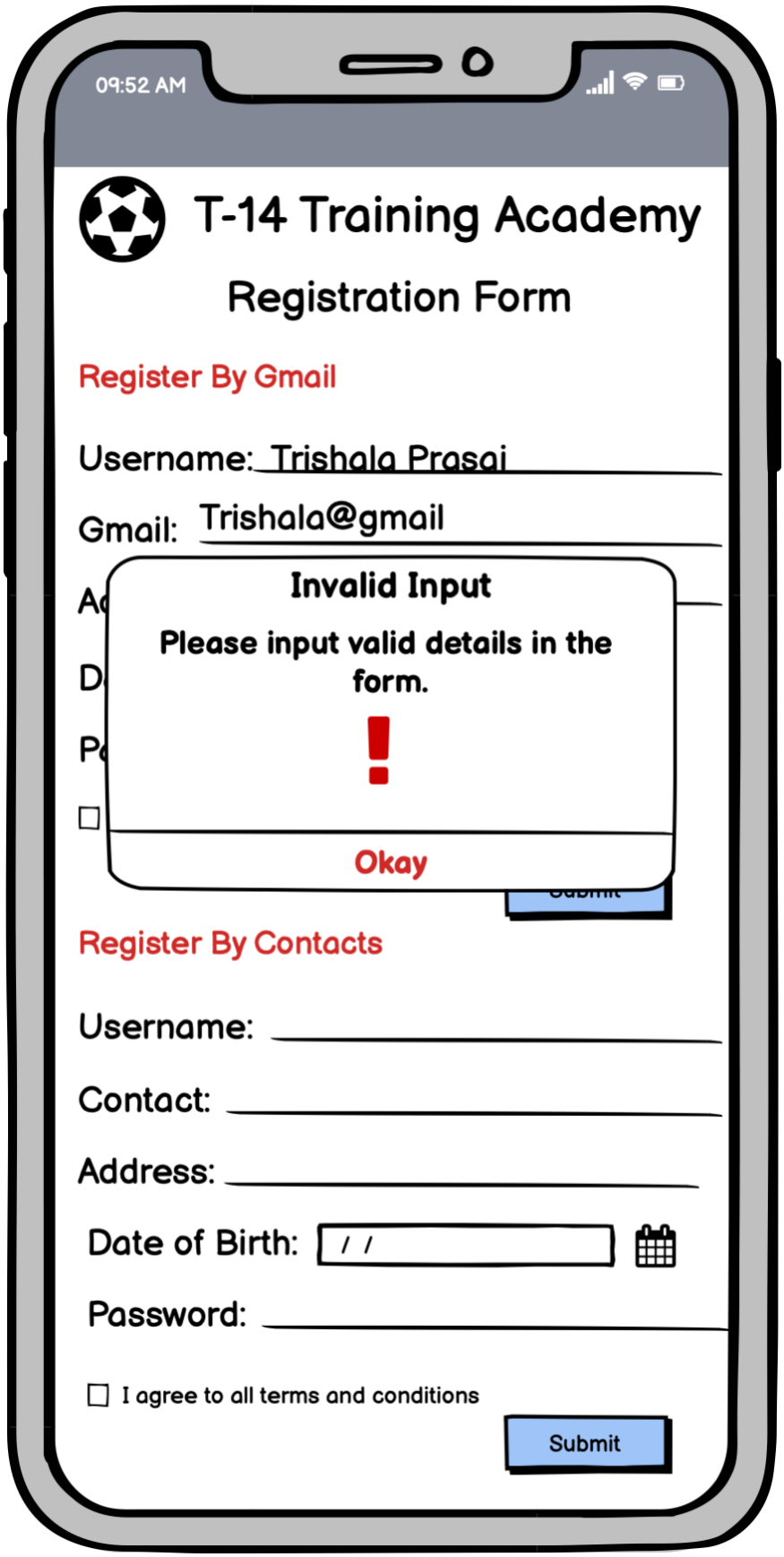
7.Prototype

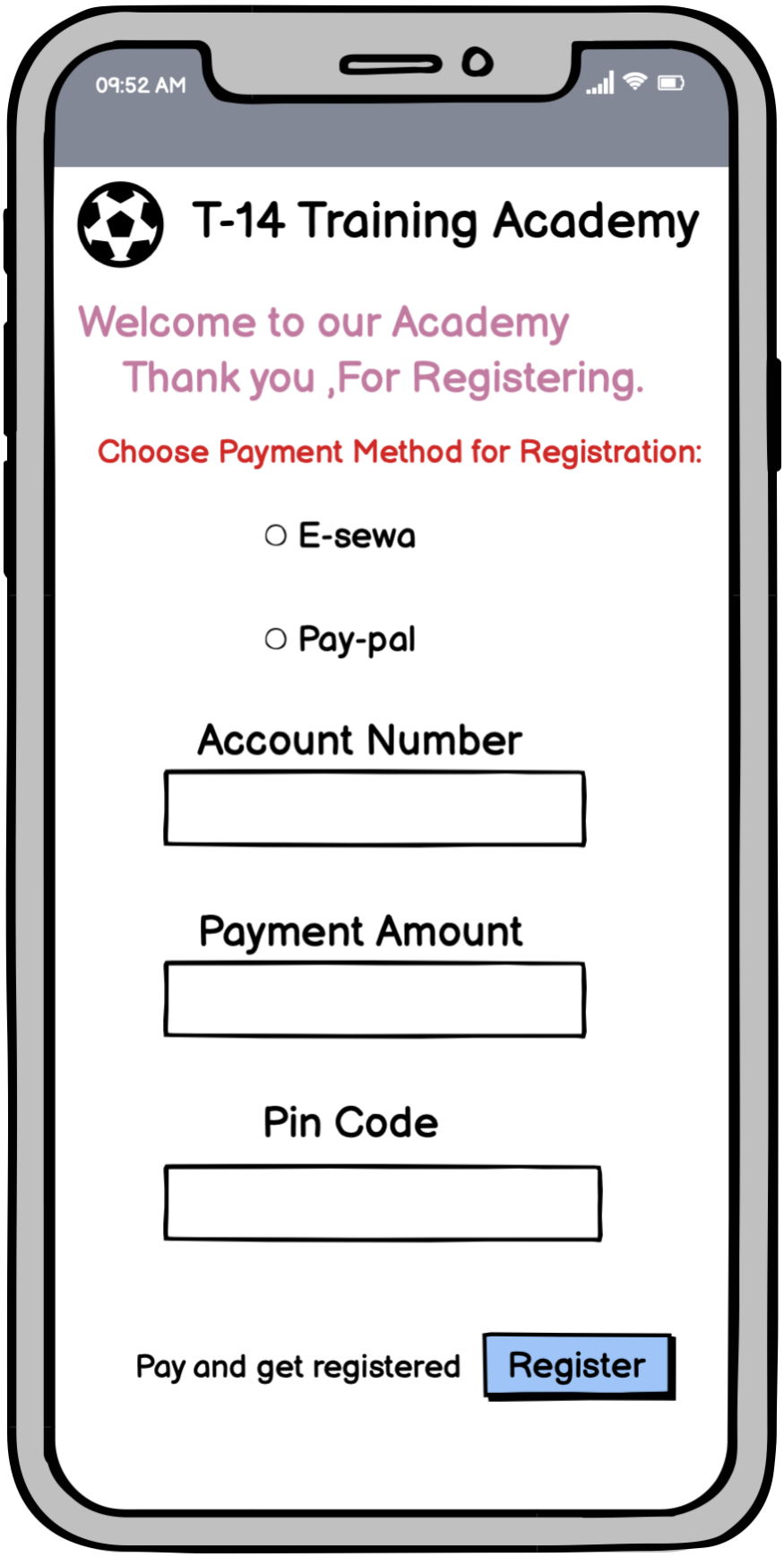

09:52 AM
T-14 Training Academy
UserName
Password
Log-In
Create a new account
Click here to register

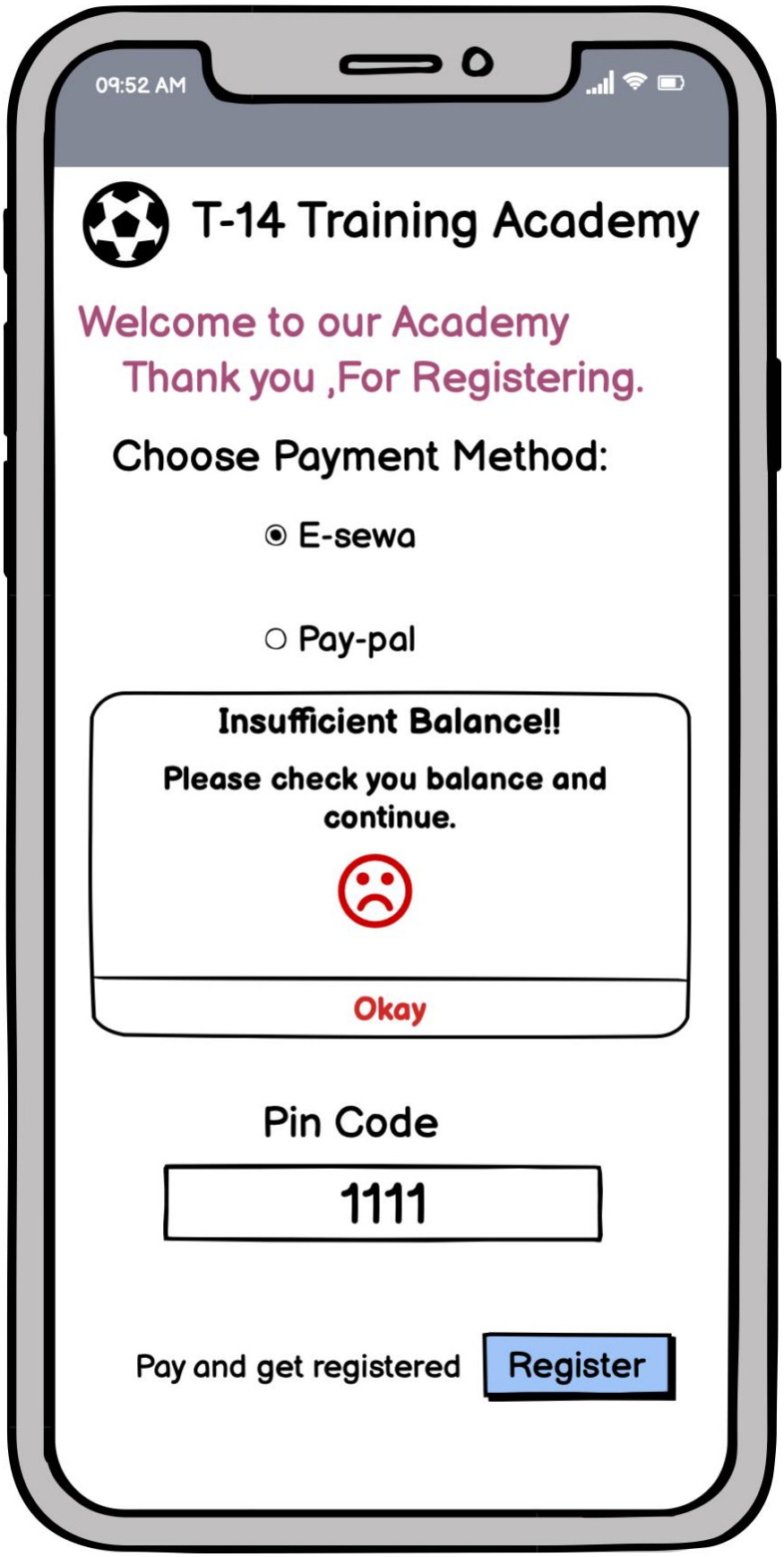

09:52 AM
T-14 Training Academy
Learn more
Profile
Registration Form
Report
Purchase Items
Objective
MCQ
Mock Test

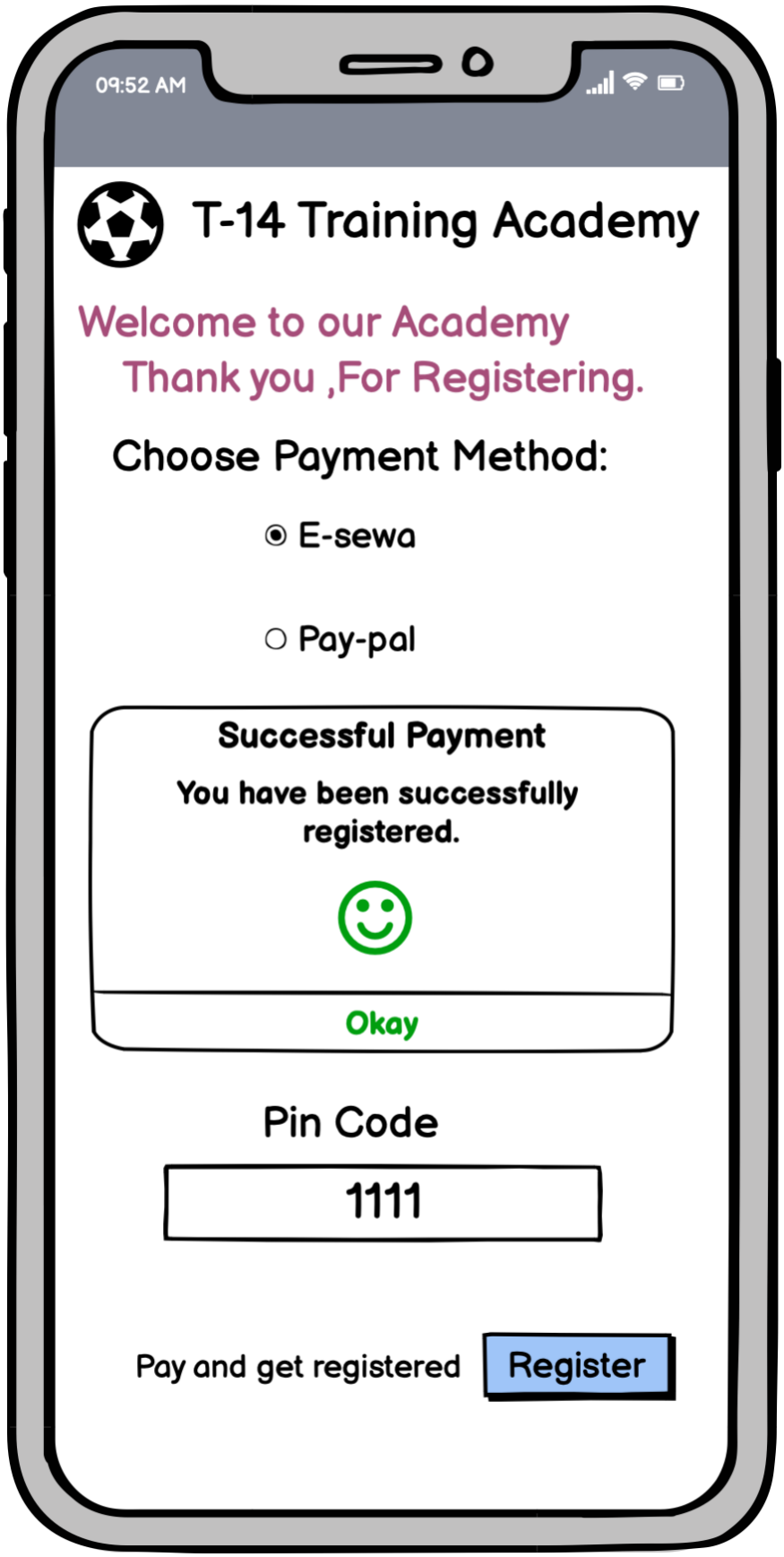

09:52 AM
T-14 Training Academy
UserName
Invalid Username / Password!!!!
Please enter valid UserName /Password.
Okay
Log-In
Create a new account
Click here to register

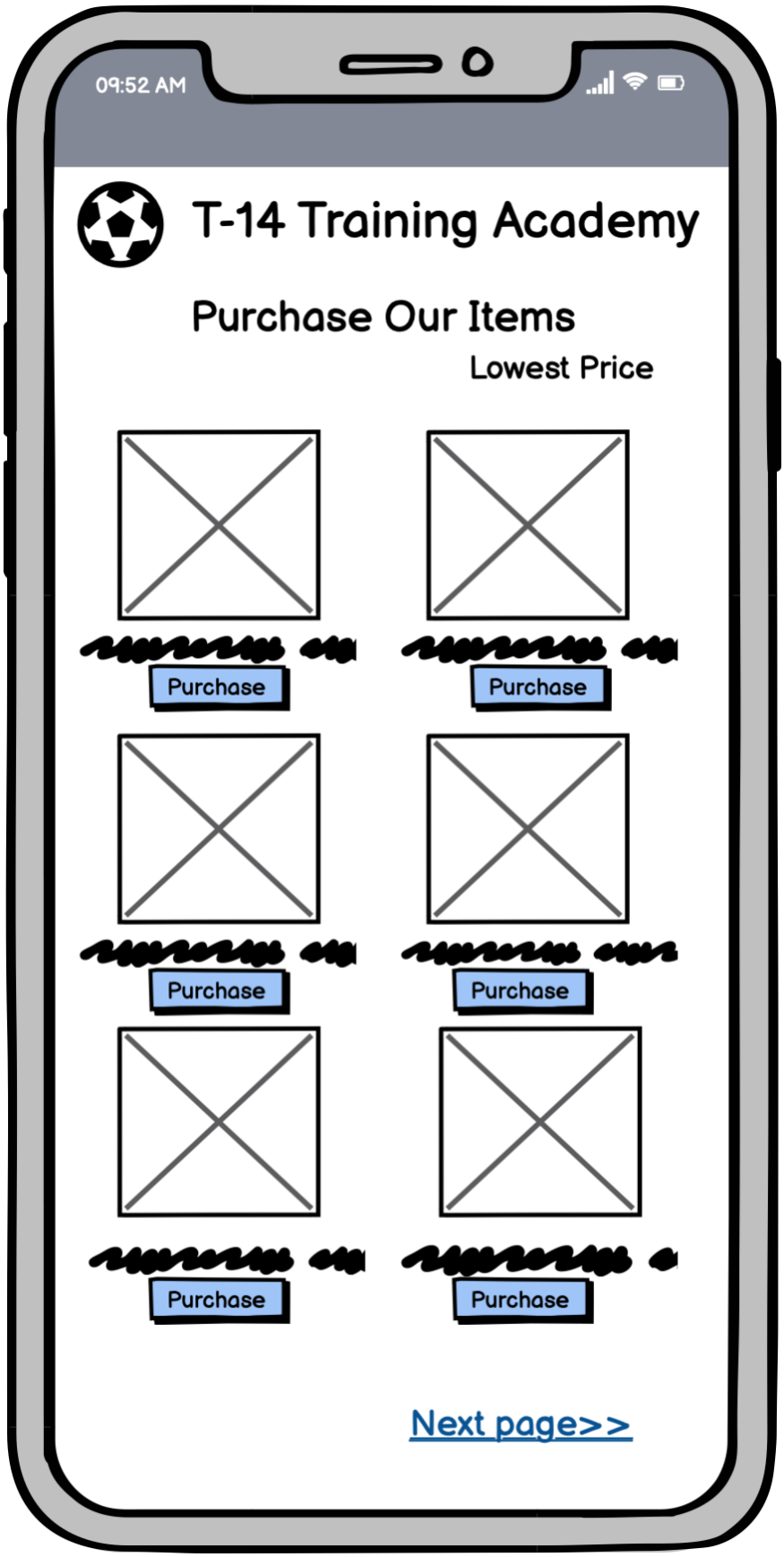

09:52 AM
T-14 Training Academy
Registration Form
Register By Gmail
Username:
Gmail:
Address:
Date of Birth:
/ /
Password:
I agree to all terms and conditions
Submit
Register By Contacts
Username:
Contact:
Address:
Date of Birth:
/ /
Password:
I agree to all terms and conditions
Submit

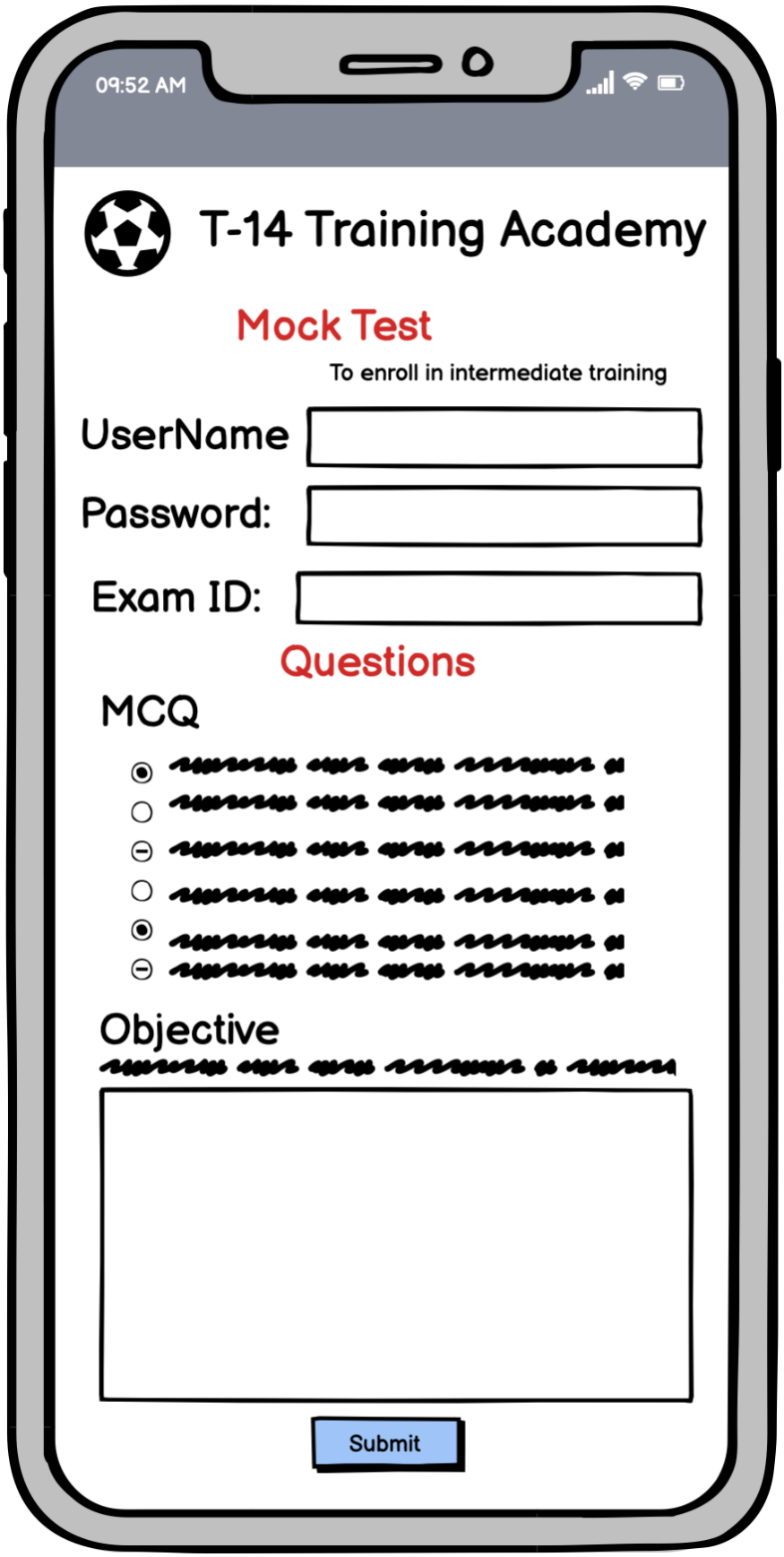

09:52 AM
T-14 Training Academy
Registration Form
Register By Gmail
Username: Trishala Prasai
Gmail: Trishala@gmail
Invalid Input
Please input valid details in the form.
Okay
Register By Contacts
Username:
Contact:
Address:
Date of Birth: / /
Password:
I agree to all terms and conditions
Submit

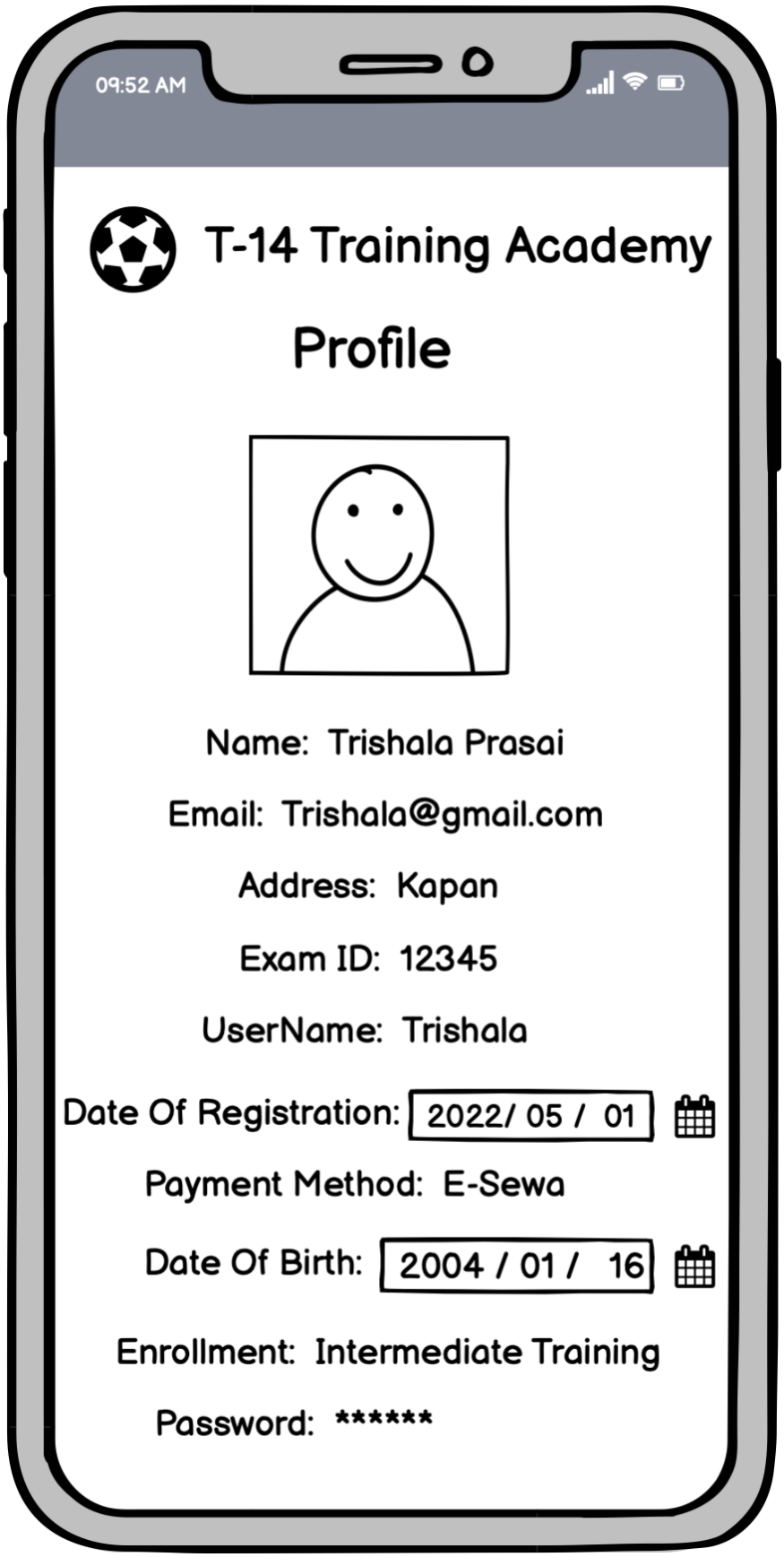

09:52 AM
T-14 Training Academy
Welcome to our Academy
Thank you ,For Registering.
Choose Payment Method for Registration:
E-sewa
Pay-pal
Account Number
Payment Amount
Pin Code
Pay and get registered
Register

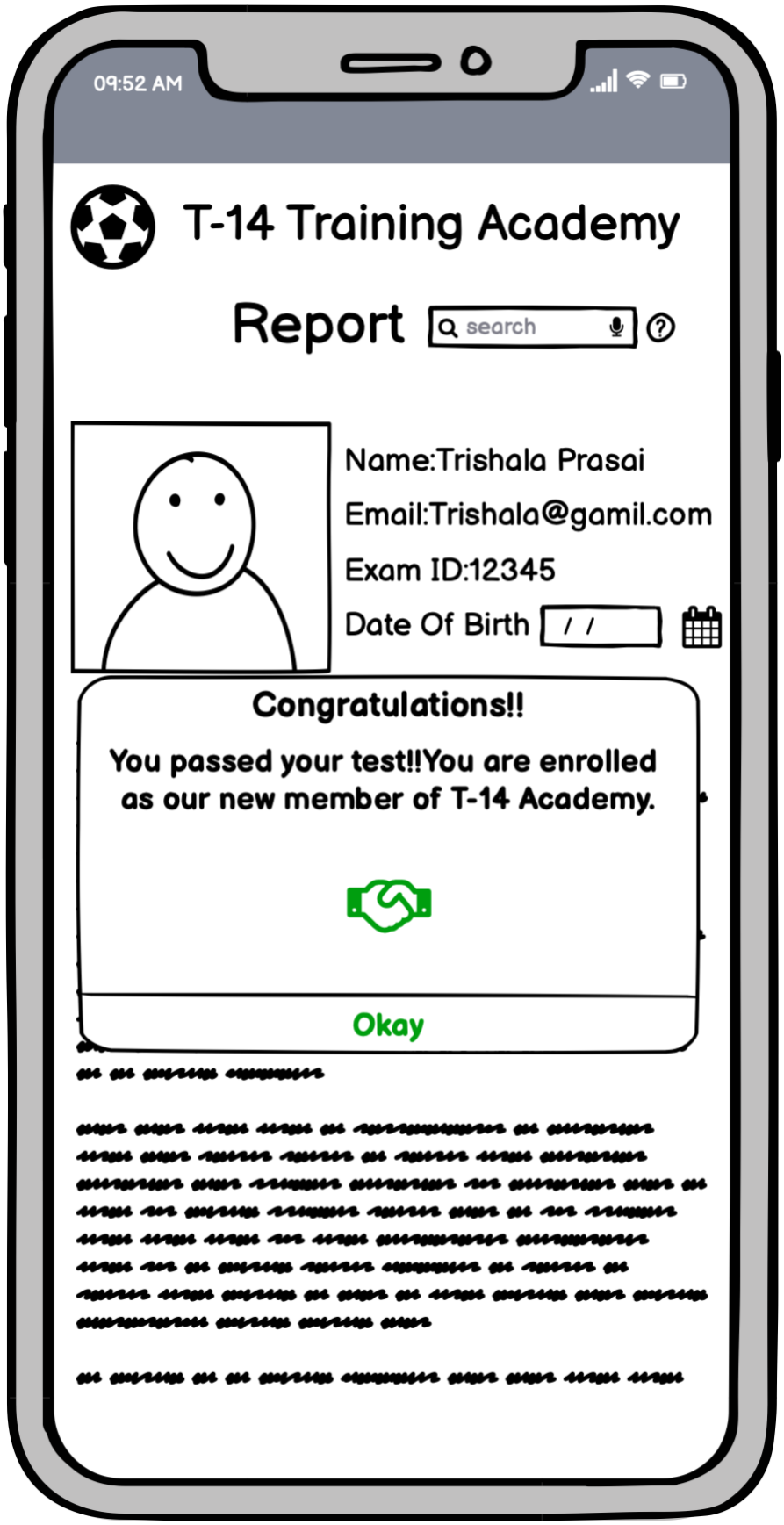

09:52 AM
T-14 Training Academy
Welcome to our Academy
Thank you ,For Registering.
Choose Payment Method:
E-sewa
Pay-pal
Insufficient Balance!!
Please check you balance and continue.
Okay
Pin Code
1111
Pay and get registered
Register

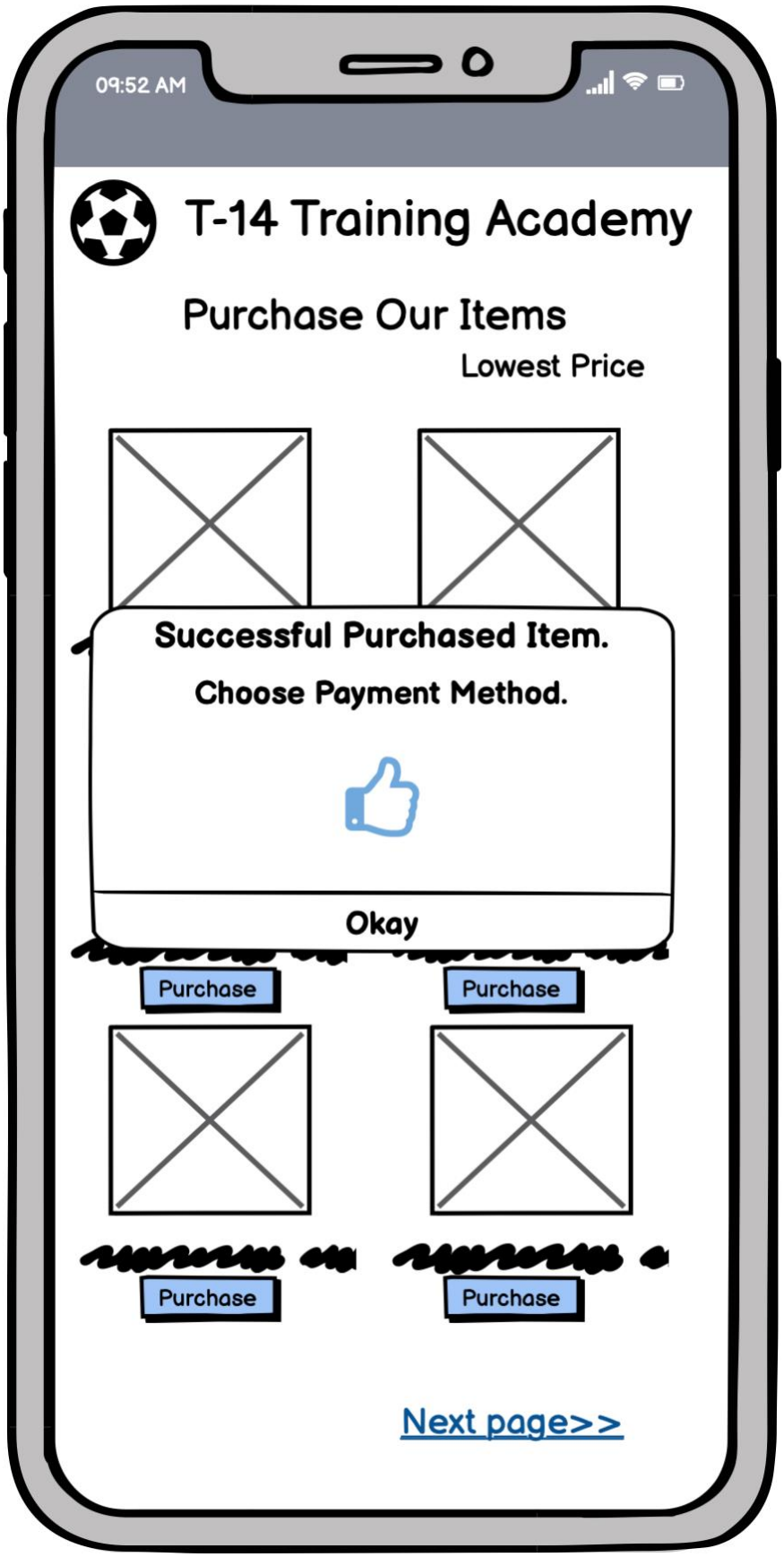

09:52 AM
T-14 Training Academy
Welcome to our Academy
Thank you ,For Registering.
Choose Payment Method:
E-sewa
Pay-pal
Successful Payment
You have been successfully registered.
Okay
Pin Code
1111
Pay and get registered
Register

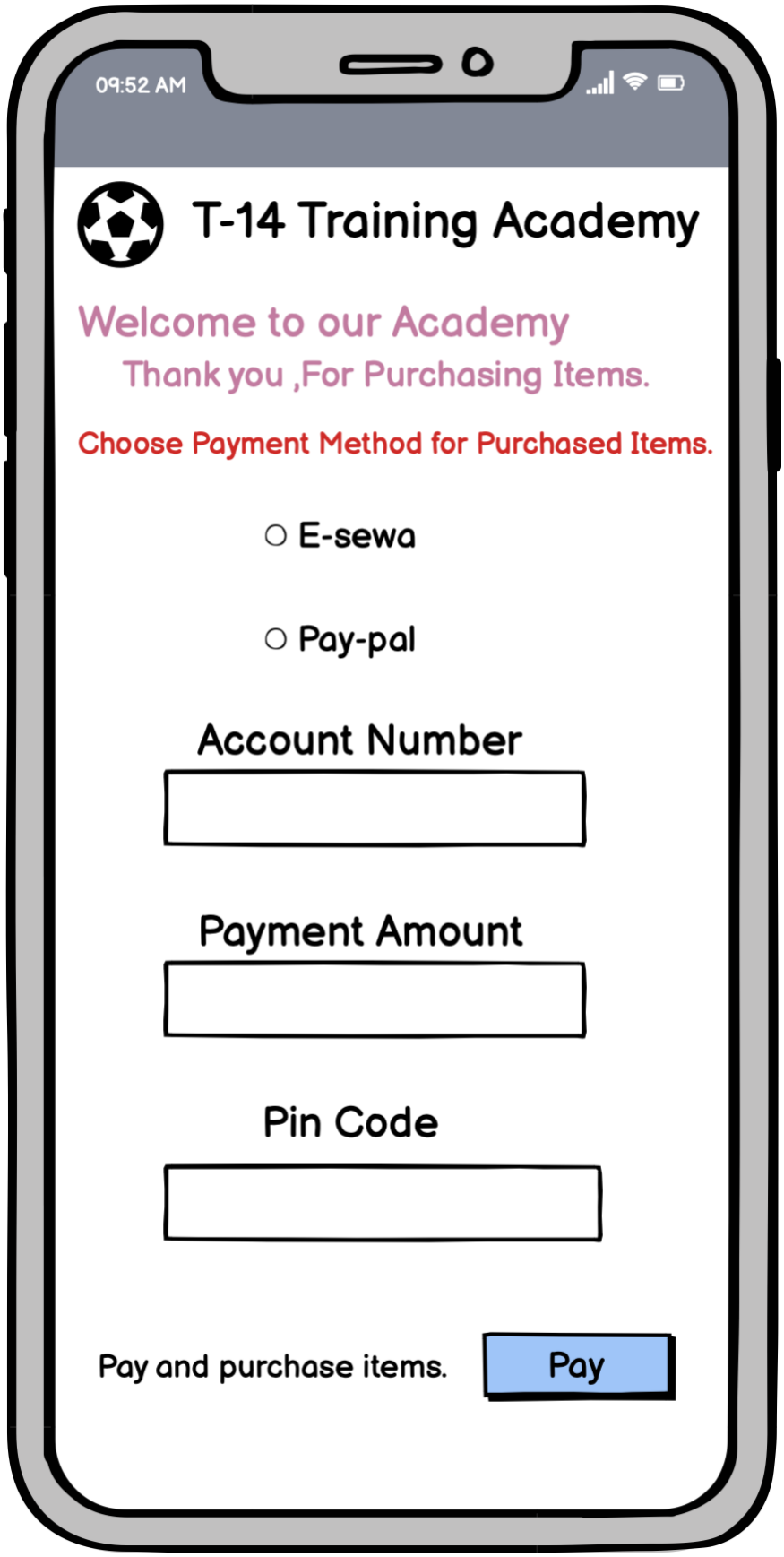

09:52 AM
T-14 Training Academy
Purchase Our Items
Lowest Price
Purchase
Purchase
Purchase
Purchase
Purchase
Purchase
Next page>>

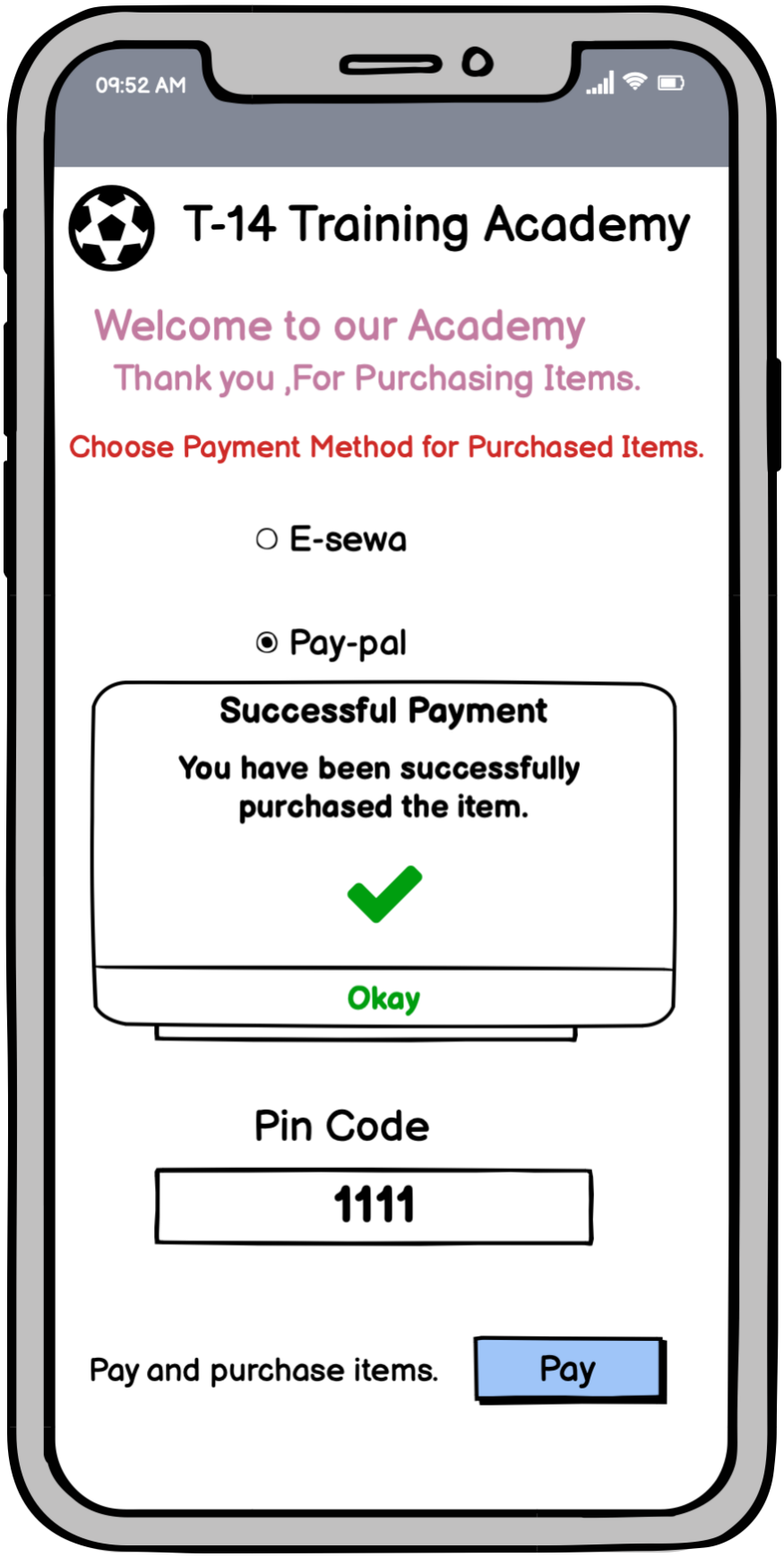

09:52 AM
T-14 Training Academy
Mock Test
To enroll in intermediate training
UserName
Password:
Exam ID:
Questions
MCQ
Objective
Submit

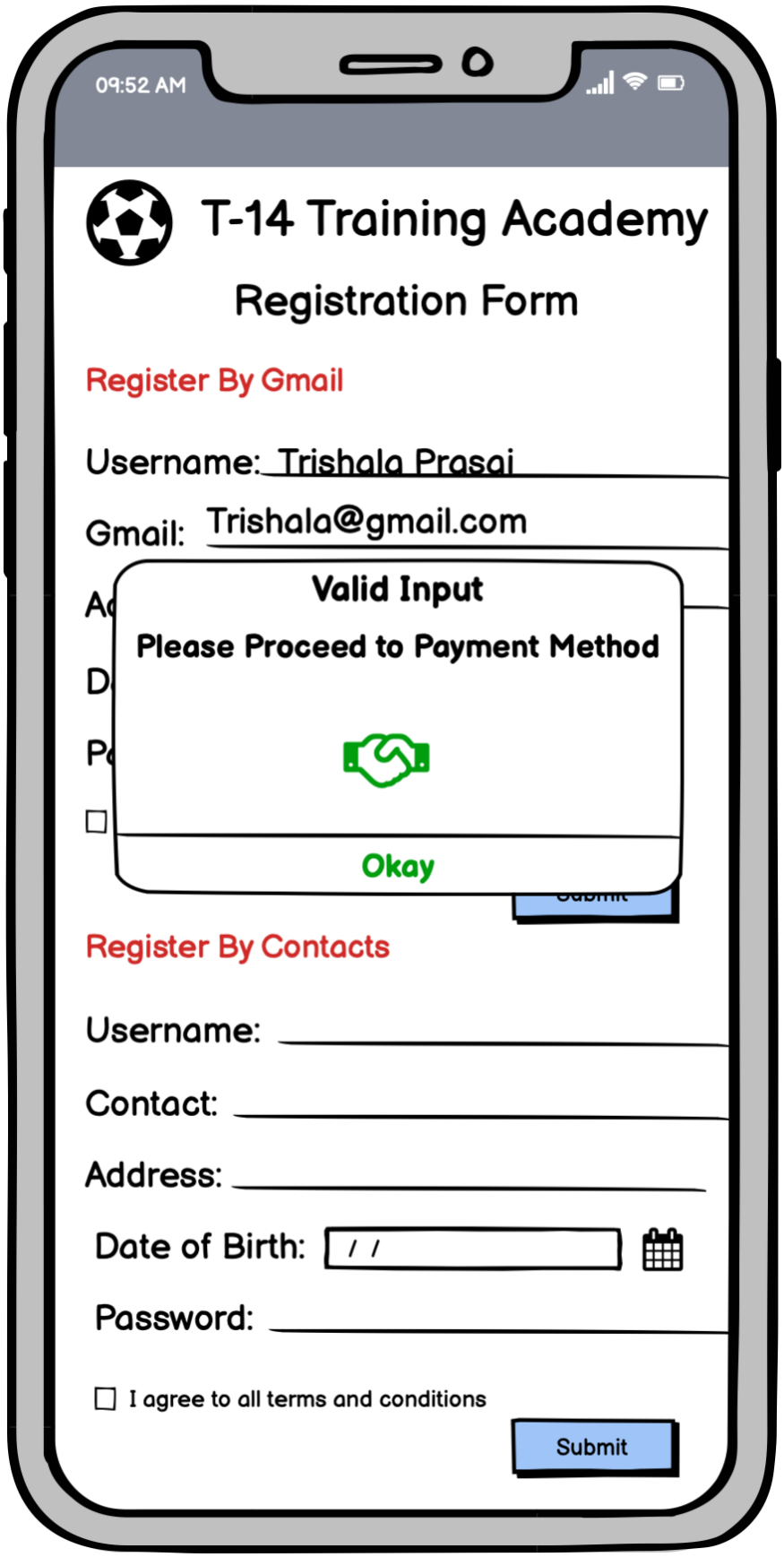

09:52 AM
T-14 Training Academy
Profile
Name: Trishala Prasai
Email: Trishala@gmail.com
Address: Kapan
Exam ID: 12345
UserName: Trishala
Date Of Registration: 2022/ 05 / 01
Payment Method: E-Sewa
Date Of Birth: 2004 / 01 / 16
Enrollment: Intermediate Training
Password: ******


09:52 AM
T-14 Training Academy
Report
search
Name:Trishala Prasai
Email:Trishala@gamil.com
Exam ID:12345
Date Of Birth
/ /
Congratulations!!
You passed your test!!You are enrolled as our new member of T-14 Academy.
Okay


09:52 AM
T-14 Training Academy
Purchase Our Items
Lowest Price
Successful Purchased Item.
Choose Payment Method.
Okay
Purchase
Purchase
Purchase
Purchase
Next page>>


09:52 AM
T-14 Training Academy
Welcome to our Academy
Thank you ,For Purchasing Items.
Choose Payment Method for Purchased Items.
E-sewa
Pay-pal
Account Number
Payment Amount
Pin Code
Pay and purchase items.
Pay


09:52 AM
T-14 Training Academy
Welcome to our Academy
Thank you ,For Purchasing Items.
Choose Payment Method for Purchased Items.
E-sewa
Pay-pal
Successful Payment
You have been successfully purchased the item.
Okay
Pin Code
1111
Pay and purchase items.
Pay


09:52 AM
T-14 Training Academy
Registration Form
Register By Gmail
Username: Trishala Prasai
Gmail: Trishala@gmail.com
Valid Input
Please Proceed to Payment Method
Okay
Register By Contacts
Username:
Contact:
Address:
Date of Birth: / /
Password:
I agree to all terms and conditions
Submit

8.Conclusion

This coursework resembles the first coursework in certain ways. This course is about object-oriented development. This coursework has resulted in the creation of a T-14Academy system. It was difficult for me to complete all of the figures and diagrams because it was all new to me.

It was a difficult task to create the sequence diagram, use case diagram, and class diagram. I never imagined I'd be able to finish all of these diagrams on time. In the development of this information system and this coursework, Gantt charts, Balsamiq for the application prototypes, draw.io for the various use case diagrams and class diagrams, and other components were used. Although complicated at first glance, with the assistance of module leaders and friends, the situation was simplified. Gantt charts are used to represent the progress of a project across time, and I learnt how to use them. It will be extremely advantageous to be able to create use cases, communication diagrams, and class diagrams. In order to clarify how the fitness software should function and what features should be included, it was necessary to create a prototype for it.

In this coursework, I learned about the procedures that must be followed in the real world before any application or website can be developed, such as when designers and programmers meet with clients and management to discuss how the application can be developed. The benefits of this will be incredibly advantageous in the long term of our future careers as Software Engineers.

Ξ

Bibliography

APM, 2022. *APM.* [Online] Available at: https://www.apm.org.uk/resources/find-a-resource/gantt-chart/#:~:text=A%20Gantt%20chart%20is%20a,useful%20for%20simplifying%20complex%20projects
[Accessed 1 May 2022].

Analysitabs, 2022. *Analysitabs.* [Online] Available at: https://www.apm.org.uk/resources/find-a-resource/gantt-chart/#:~:text=A%20Gantt%20chart%20is%20a,useful%20for%20simplifying%20complex%20projects
[Accessed 1 May 2022].

Tech Target, 2022. *Collabaration Diagram.* [Online] Available at: https://www.techtarget.com/searchsoftwarequality/definition/collaboration-diagram#:~:text=A%20collaboration%20diagram%2C%20also%20known,the%20role%20of%20each%20object
[Accessed 2022].

The qalead, 2022. *dsdm.* [Online] Available at: https://theqalead.com/topics/dsdm-dynamic-systems-development-method/
[Accessed 2022].

Tutorials point, 2022. *MVC framework.* [Online] Available at: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
[Accessed 2022].

performance syllabus, 2022. *software testing.* [Online] Available at: https://performancelabus.com/software-testing-importance-sdlc/#:~:text=Testing%20Phase%20in%20SDLC&text=The%20main%20goal%20of%20the,how%20they%20affect%20each%20other.
[Accessed 2022].

Thalesgroup, 2022. *4 types of software maintainance.* [Online] Available at: https://cpl.thalesgroup.com/software-monetization/four-types-of-software-maintenance
[Accessed 2022].