



**Daffodil**  
*International*  
**University**

**SHIKHUN**

Learn new skills online

This project report has been submitted in fulfillment of the requirements for the Degree of  
Bachelor of Science in Software Engineering

**Submitted By**

**Md. Amin Hossain**

**ID: 172-35-2180**

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

**Supervised By**

**Md. Anwar Hossen**

**Assistant Professor**

Department of Software Engineering

Faculty of Science and Information Technology

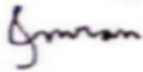
Daffodil International University

## APPROVAL

---

This Project titled “**Shikun**”, submitted by **Md. Amin Hossain, ID: 172-35-2180** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

### BOARD OF EXAMINERS



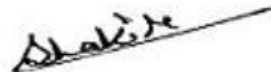
-----  
Dr. Imran Mahmud  
Associate Professor and Head  
Department of Software Engineering  
Daffodil International University

Chairman



-----  
Md Anwar Hossen  
Assistant Professor  
Department of Software Engineering  
Daffodil International University

Internal Examiner 1



-----  
Asif Khan Shakir  
Senior Lecturer  
Department of Software Engineering  
Daffodil International University

Internal Examiner 2



-----  
Professor Dr M Shamim Kaiser  
Institute of Information Technology  
Jahangirnagar University

External Examiner

## DECLARATION

---

The project entitled “**Shikhun**” is done under the supervision of **Md. Anwar Hossen, Assistant Professor** and Department of Software Engineering of Daffodil International University. I declare that this project is my original work for the degree of B.Sc. in Software Engineering and that neither the whole work nor any part has been submitted for another degree at any other university.

### Submitted By

*Amin Hossain*

-----

**Md. Amin Hossain**

ID: 172-35-2180

Batch: 23<sup>th</sup>

Department of Software Engineering

Daffodil International University

### Supervised By

  
30.1.21

-----

**Md. Anwar Hossen**

Assistant Professor

Department of Software Engineering

Daffodil International University

## ACKNOWLEDGEMENT

---

First of all, I'm really grateful to the Almighty Allah for making me eligible and giving me the patience to complete the project. I would also like to thank my supervisor **Mr. Md. Anwar Hossen**, Assistant professor, Department of Software Engineering. I'm extremely grateful and indebted to him as he has given his expert, sincere and valuable guidance and encouragement to successfully complete the project.

I would like to thank the HRDI Department of Daffodil International University for believing in me and their continuous support and engagement. Without their passionate participation and input, the project would not be possible to conduct successfully. We take this occasion to convey our sincere thanks to the Honorable **Head of the Department** and all **Faculty Members** of the Department of Software Engineering and specially our supervisor **Mr. Md. Anwar Hossain** for their help and encouragement.

## EXECUTIVE SUMMARY

---

**Shikhun** is an online learning platform where learners will connect with the experts of different stacks to acquire a new skill through some advanced techniques and expert guidance. This new skill will help them to prepare for the competitive job market or the next big story they want to create.

The main purpose of “**Shikhun**” is described more in the first chapter. By going through it one can learn the background, motivation, objective, benefits and goals, stakeholders of the project. One can also get the rough idea by going through the system model in this section.

The second chapter contains the feasibility study of the project. Where I discussed different feasibility consequences of the project.

The third chapter is about project requirements. These requirements were needed to be fulfilled by the application before coming to live. These are mainly the functional requirements and non-functional requirements.

Four, Five, Six, Seven and Eight chapters are respectively about projecting different levels of system diagrams like Use Case Diagram, Activity Diagram, Sequence Diagram, ERD Diagram and Class Diagram.

Stockholders play a big part in this system. So, the user interface is created in such a way which is very easy to understand. Admin also plays a big role. So, maximum maintainability is given to the Admin panel.

Developing is not the end of a project. It has to work perfectly before we can call it a functional project. For this, we have to make sure the quality of the product by undergoing various test cases. The test cases also help to find out the vulnerabilities of the system. Also helps to remove the vulnerabilities and fix the bugs before the release date.

The system UI chapter acts as a guide for the user. It helps them to get acquainted with the system easily. It contains all the information on how to browse the application properly.

**NOTE:** This project is an industry collaboration between Department of Software Engineering and Human Resource Development Institute (HRDI) wing of Daffodil International University.

## TABLE OF CONTENTS

---

### Table of Contents

<b>APPROVAL .....</b>	<b>i</b>
<b>DECLARATION.....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iii</b>
<b>EXECUTIVE SUMMARY .....</b>	<b>iv</b>
<b>TABLE OF CONTENTS .....</b>	<b>v-vii</b>
<b>LIST OF FIGURES .....</b>	<b>viii</b>
<b>Chapter 1: Introduction .....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Objective .....	1
1.3 Project Overview .....	1
1.4 Project Purpose .....	1
1.4.1 Background .....	1
1.4.2 Benefits and Beneficiaries .....	2
1.4.3 Goals .....	2
1.5 Stakeholders.....	2
1.5.1 Guest .....	2
1.5.2 Learner .....	2
1.5.3 Trainer.....	2
1.5.4 Customer.....	3
1.5.5 Admin .....	3
1.6 Features & Application Structure .....	3
1.7 System Architecture.....	4
1.8 Project Schedule.....	4
<b>Chapter 2: System Analysis .....</b>	<b>5</b>
2.1 Feasibility Study .....	5
2.1.1 Technical Feasibility.....	5
2.1.2 Operational Feasibility.....	5
2.1.3 Economical Feasibility.....	5
2.1.4 Legal Feasibility.....	5
2.1.5 Scheduling Feasibility.....	5

<b>Chapter 3: Software Requirement Specification .....</b>	<b>6</b>
3.1 Functional Requirements .....	6-7
3.2 Non-Functional Requirements .....	8
3.2.1 Performance .....	8
3.2.2 Safety & Security Requirements.....	8
3.2.3 Reliability & Availability .....	8
3.2.4 Maintainability and Supportability .....	8
<b>Chapter 4: Use Case Diagram and Description .....</b>	<b>9</b>
4.1 Registration Use Case .....	9
4.1.1 Registration Use Case Description .....	10
4.2 Course Implementation.....	11
4.2.1 Course Use Case Description.....	12
4.3 Lesson Implementation.....	13
4.3.1 Lesson Use Case Description.....	14
4.4 Summit Implemenattion.....	15
4.4.1 Summit Use Case Description .....	16
4.5 Live Evenets Implementation .....	17
4.5.1 Live Events Use Case Description.....	18
4.6 Merchandise Product Implementation .....	19
4.6.1 Merchandise Product Use Case Description.....	20
4.7 Manage System Implementation.....	21
4.7.1 Manage System Use Case Description .....	22
<b>Chapter 5: Activity Diagram .....</b>	<b>23</b>
5.1 Registration Activity Diagram .....	23
5.2 Course Implementation Activity Diagram.....	24
5.3 Lesson Implementation Activity Diagram.....	25
5.4 Summit Implementation Activity Diagram .....	26
5.5 Live Event Implementation Activity Diagram .....	27
5.6 Merchandise Product Implementation Activity Diagram .....	28
5.7 Manage System Activity Diagram.....	29

<b>Chapter 6: Sequence Diagram</b> .....	<b>30</b>
6.1 Registration Sequence Diagram.....	30
6.2 Course Implementation Sequence Diagram for Trainer & Admin.....	31
6.3 Course Implementation Sequence Diagram for Learner.....	32
6.4 Lesson Implementation Sequence Diagram for Trainer & Admin.....	33
6.5 Lesson Implementation Sequence Diagram for Learner.....	34
6.6 Summit Implementation Sequence Diagram for Trainer & Admin.....	35
6.7 Summit Implementation Sequence Diagram for Learner.....	36
6.8 Live Event Implementation Sequence Diagram for Trainer & Admin.....	37
6.9 Live Event Implementation Sequence Diagram for Learner.....	38
6.10 Merchandise Implementation Sequence Diagram for Admin.....	39
6.11 Merchandise Implementation Sequence Diagram for Customer.....	40
6.12 Manage System Sequence Diagram.....	41
<b>Chapter 7: Shikhun’s Entity Relationship Diagram(ERD)</b> .....	<b>42</b>
<b>Chapter 8: Class Diagram</b> .....	<b>43</b>
8.1 Shikhun’s Class Diagram.....	43
8.2 Detail Desingn of Classes.....	44-52
<b>Chapter 9: System Testing</b> .....	<b>53-58</b>
<b>Chapter 10: User Manual</b> .....	<b>59-65</b>
<b>Chapter 11: Tools &amp; Technologies</b> .....	<b>66</b>
<b>Chapter 12: Project Important Links</b> .....	<b>66</b>
<b>Chapter 13: Project Summary</b> .....	<b>67</b>
13.1 Limitations.....	67
13.2 Obstacles & Achieevements.....	67
13.3 Future Scope.....	67
<b>REFERENCES</b> .....	<b>68</b>
<b>PLAGIARISM REPORT</b> .....	<b>69</b>
<b>ACCOUNCE CLEARANCE</b> .....	<b>70</b>



## LIST OF FIGURES

Figure No.		Page
1.7	Shikhun's System Architecture	4
1.8.1	Gantt Chart	4
4.1	Registration Use Case Diagram	9
4.2	Course Implementation Use Case Diagram	11
4.3	Lesson Implementation Use Case Diagram	13
4.4	Summit Implementation Use Case Diagram	15
4.5	Live Event Implementation Use Case Diagram	19
4.6	Merchandise Product Use Case Diagram	19
4.7	Manage System Use Case Diagram	21
5.1	Registration Activity Diagram	23
5.2	Course Implementation Diagram	24
5.3	Lesson Implementation Diagram	25
5.4	Summit Implementation Diagram	26
5.5	Live Event Implementation Diagram	27
5.6	Merchandise Activity Diagram	28
5.7	Manage Activity Diagram	29
6.1	Registration Sequence Diagram	30
6.2	Course Sequence Diagram for Admin & Trainer	31
6.3	Course Sequence Diagram for Learner	32
6.4	Lesson Implementation Sequence Diagram for Admin & Trainer	33
6.5	Lesson Implementation Sequence Diagram for Learner	34
6.6	Summit Implementation Sequence Diagram for Admin & Trainer	35
6.7	Summit Implementation Sequence Diagram for Learner	36
6.8	Live Event Implementation Sequence Diagram for Admin & Trainer	37
6.9	Live Event Implementation Sequence Diagram for Learner	38
6.10	Merchandise Product Sequence Diagram for Admin	39
6.11	Merchandise Product Sequence Diagram for Customer	40
6.12	Manage System Sequence Diagram	41
7.1	Shikhun's ER Diagram	42
8.1	Shikhun's Class Diagram	43

### 1.1 Motivation

Nowadays skills are an important part of our professional life and career. To acquire a skill we need time and cost both at the same time. But most of the time it seems hard to manage the time physically where the cost is flexible, as a result people move to online learning platforms to acquire their desired skills. At this point Daffodil International University HRDI wing introduced the ‘**Shikhun**’ online learning platform where people can grow their skill with the most advanced techniques. And also people will guide under the experts of different stacks.

### 1.2 Objective

The objective of ‘**Shikhun**’ is connecting the experts of different stacks with the learners through an online platform. Where a competitive and automated process will help the experts and the learners to ready themselves for the competitive job market and also go ahead of their career goal.

### 1.3 Project Overview

Shikhun is a blended online learning platform. It will help people to grow their skill under the guidance of experts of different stacks and advanced techniques. It will provide 24/7 online access to the platform to continue learning from anywhere and anytime. It will be available for Web and Mobile both platforms. It will be built with all the latest technologies. Shikhun initially will be released for the web.

### 1.4 Project Purpose

When we start learning something new most of the time we face many problems to complete our learning process. ‘**Shikhun**’ will provide well-organized and expert trainers to the learners to solve their problems and achieve their desired skill and goal through the application 24/7 from anywhere and anytime.

#### 1.4.1 Background

We are living in an era of modern technologies. And for that we have to be up-to-date with different skills or expertise. As a result, most of the time people try to utilize their time in learning something new to enhance their skill. But sometimes it’s difficult to manage the learning time physically. As a result people like to enhance their desired skill through online platforms. ‘**Shikhun**’ offers the most effective solution regarding those situations.

### 1.4.2 Benefits and Beneficiaries

‘**Shikhun**’ introduces the best solution to the online learners to grow their skills through a web platform 24/7 from anywhere and anytime. Some benefits and beneficiaries mentioned below:

- 24/7 access from anywhere and anytime
- Expert trainers for different stacks
- Personalized automated dashboard
- Experts suggestion according to profile & progress
- Competitive environment to boost the learning progress
- Chance to join Webinar, Events, Podcasts on latest and trendy topics

### 1.4.3 Goals

‘**Shikhun**’ have several goals mentioned below –

- 24/7 access from anywhere and anytime to continue learning.
- Connect the experts with the learners
- Solid career guide line according to the learner interest
- Provide Webinar, Events, Podcasts on latest and trendy topics

## 1.5 Stockholders

### 1.5.1 Guests

In ‘**Shikhun**’ the Guests will play minimal role. They can do minimal activities as an unregistered member of the system. They can browse courses, join free webinar, free events, free podcasts and can read blogs.

### 1.5.2 Learners

In ‘**Shikhun**’ the Learners role will need registration to access the learner’s modules. Learners will have all learner modules and content access. They will have privileges to personalize a dashboard, free courses and lessons, paid courses and lessons by purchase, paid and free webinar, events, podcasts, premium blogs, connect with other learners, interact with expert’s trainer and many more.

### 1.5.3 Trainers

In ‘**Shikhun**’ the Trainers role must need registration to get trainer privileges. Trainer can add free and paid courses, arrange free and paid webinars, events, and podcasts. Can manage Learners, post blogs, personalized dashboard and many more.

### **1.5.4 Customers**

In ‘**Shikhun**’ the customer will play a minimal role. They will be able to browse products, purchase products, can manage purchase, account etc.

### **1.5.5 Admin**

In ‘**Shikhun**’ Admin will have all types of access control for all the modules. Can manage Learners, Trainers, System and many more.

## **1.6. Features & Application Structure**

‘**Shikhun**’ will have many features and a user-friendly interface for the users.

### 1.6.1 Pages

Home, Courses (online, offline), Merchandise, Events (summit, live event, webinar, podcast), About, Team, FAQ, Contact etc.

### 1.6.2 Functionality

- Registration, Login, Authentication
- Create course, lessons, quizzes, webinar, live sessions, podcasts, summits
- Enroll courses, Participate Courses, Webinar, Live Session, Podcasts, Summits
- Complete Courses, Get Certificate
- Add/Update/Delete merchandise products, sell merchandise products
- Manage application settings, Dashboard Personalization, Generate suggestion, Notification etc.

## 1.7 Shikhun's System Architecture

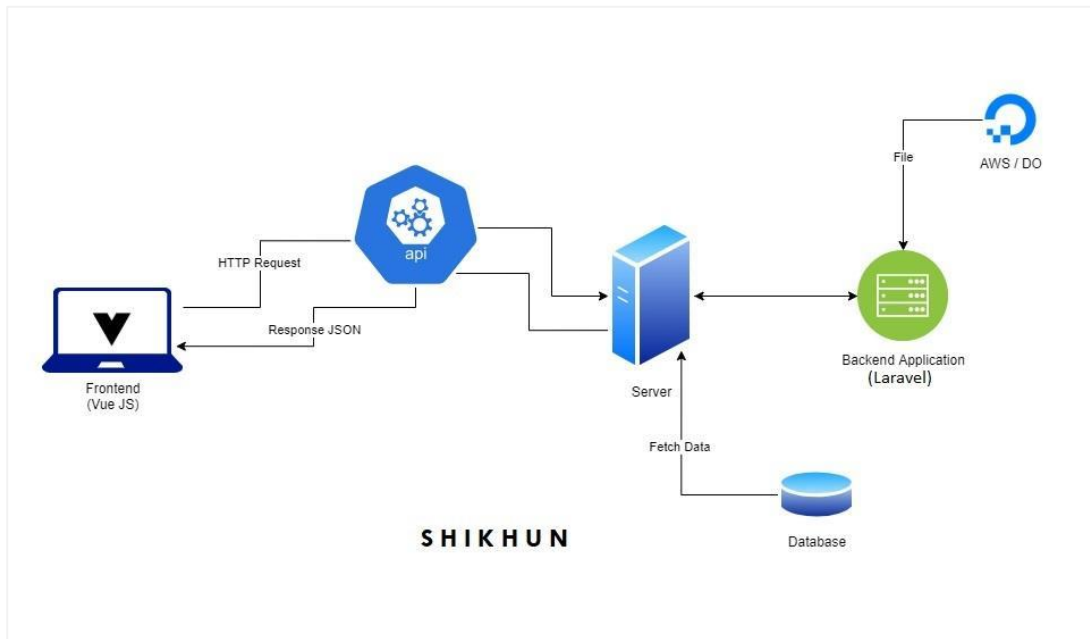


Fig 1.7: Shikhun System Architecture

## 1.8 Project Schedule

### 1.8.1 Gantt Chart

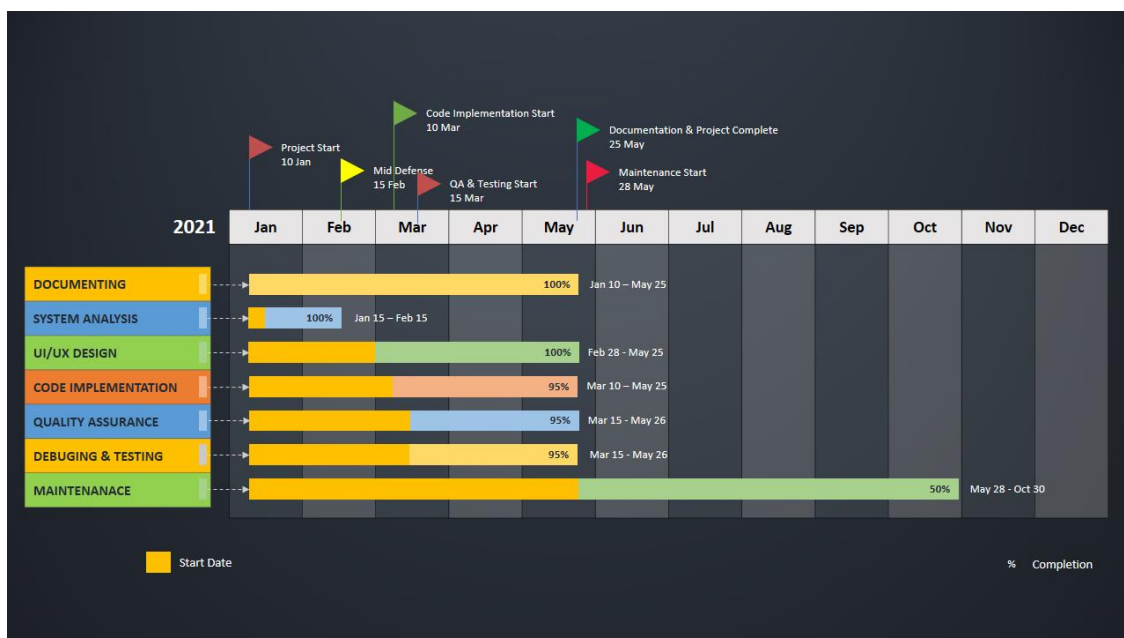


Fig 1.8.1: Gantt chart

### 2.1 Feasibility Study

In this software project's feasibility, we will be looking over all possible relevant factors like economical, technical, operational, legal and scheduling feasibility. Throughout this feasibility study we will be able to make a decision that our team is capable enough to develop this project or not. We will be analyzing those feasibility by analyzing the tools and technologies we are going to use and their requirements we are finding to have in our team members to contribute in different portions of this project are summarized below.

#### 2.1.1 Technical Feasibility

We are using PHP (Laravel Framework), MySQL (database), JavaScript, Vue.js (JavaScript Framework), Bootstrap, HTML5, CSS3 etc. trendy web technologies for developing 'Shikhun'. As two team members are involved with the project we need two workable PCs. We are using Windows 10 Pro as an operating system on each PC. Also a database server is needed to store data in a secure way. We have sufficient technical expertise to convert our ideas into working systems. Hopefully, we will be able to meet our technical capacity.

#### 2.1.2 Economical Feasibility

Economic feasibility is a kind of cost-benefit analysis of this project. After a cost-benefit analysis of 'Shikhun' we found that it is possible to implement. Economic feasibility deals with how much cost we need to develop this project additionally with how much production cost we need. We will analyze this part more in the cost-analysis section of this chapter.

#### 2.1.3 Legal Feasibility:

Any part of the project doesn't go against the law of our country. We have analyzed whether our project goes against data protection and acts as data storage playing a big role. Also this project is legally publishable and users can use the system without any hesitation.

#### 2.1.4 Scheduling Feasibility

The project has been scheduled for 4 months. The project is scheduled in such a way that from requirements gathering to implementation, testing maximum possible time has been assigned. We believe within this timeline we'll be able to release a beta version for users. We will analyze this part more in the time-analysis section of this chapter.

## Chapter 3: Software Requirements Specifications

---

### 3.1 Functional Requirements

<b>FR-01</b>	Role based user Registration
<b>Description</b>	Learners, Trainers will registered by own Admin will have secret access privileges
<b>Stakeholders</b>	Learners, Trainers, Admin
<b>Priority</b>	Admin & Trainers = High, Learners = Low

<b>FR-02</b>	Courses Implementation (Free + Paid)
<b>Description</b>	Trainers can add, view, update, delete courses Learners can browse, purchase, enroll, complete courses, Receive certificate after successful course completion and also can give feedback.
<b>Stakeholders</b>	Trainers, Learners
<b>Priority</b>	Trainers = High and Learners = Low

<b>FR-03</b>	Lesson Implementation
<b>Description</b>	Trainers can add, view, update, delete lessons Learners can complete lessons,
<b>Stakeholders</b>	Trainers, Learners
<b>Priority</b>	Trainers = High and Learners = Low

<b>FR-04</b>	Summit (Free + paid)
<b>Description</b>	Admin & Trainers can announce and host summit Learners can join summit
<b>Stakeholders</b>	Trainers, Learners
<b>Priority</b>	Admin & Trainers = High, Learners = Low

<b>FR-05</b>	Live Events Implementation(Free + paid)
<b>Description</b>	Trainers can announce and host live events Learners can join events
<b>Stakeholders</b>	Trainers, Learners
<b>Priority</b>	Trainers = High, Learners = Low

<b>FR-06</b>	Merchandise Products Implementation
<b>Description</b>	Merchandise can add/update/delete books, journals, badges, T-Shirts for sale. Customer can purchase the merchandise products
<b>Stakeholders</b>	Merchandise, Customer
<b>Priority</b>	Low

<b>FR-07</b>	Manage System
<b>Description</b>	Admin will be able to manage all type of activates and module of the system and block or unblock the users, courses review etc.
<b>Stakeholders</b>	Admin
<b>Priority</b>	High



## 3.2 Non-Functional Requirements

### 3.2.1 Performance

<b>FR-01</b>	All page will response or load faster
<b>Description</b>	Response to view information shall take no longer than 3 seconds to appear on the screen
<b>Stakeholders</b>	Learners, Trainers, Admin
<b>Priority</b>	High

### 3.2.2 Safety & Security

<b>FR-02</b>	All data should be in a secure mode.
<b>Description</b>	System will be designed in secure mode so that there is no chance to leak or damage any data.
<b>Stakeholders</b>	Learners, Trainers, Admin, Developers
<b>Priority</b>	High

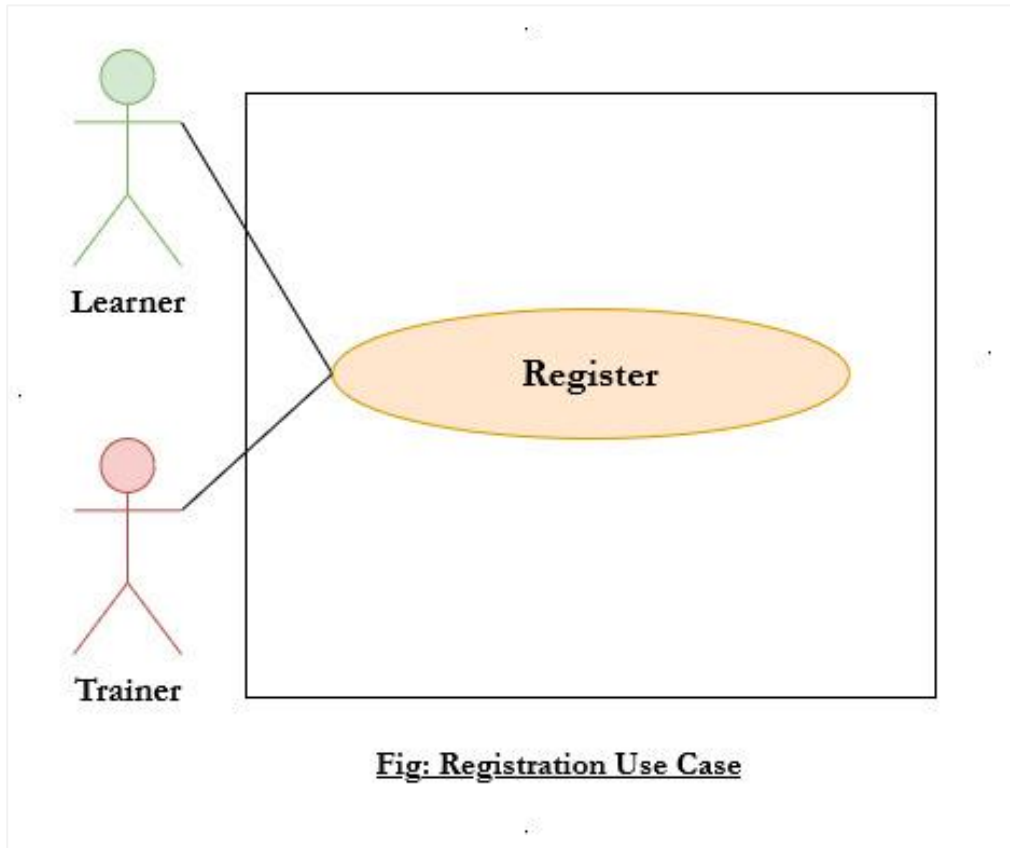
### 3.2.3 Reliability & Availability

<b>FR-03</b>	The system must be available 24/7
<b>Description</b>	The system will need a trusted and good server so that uptime will respond 24/7 as close as possible.
<b>Stakeholders</b>	Guests, Learners, Trainers, Admin
<b>Priority</b>	High

### 3.2.4 Maintainability and Supportability

<b>FR-04</b>	System will always maintain according to needs
<b>Description</b>	The system is up-to-date according to timeline and needs. Bug fix will be counted as zero tolerance and fix as soon as possible
<b>Stakeholders</b>	Admin and Developer
<b>Priority</b>	High

### 4.1 Registration Use Case Diagram



**NOTE:** For better view please follow the link – [Registration Use Case](#)

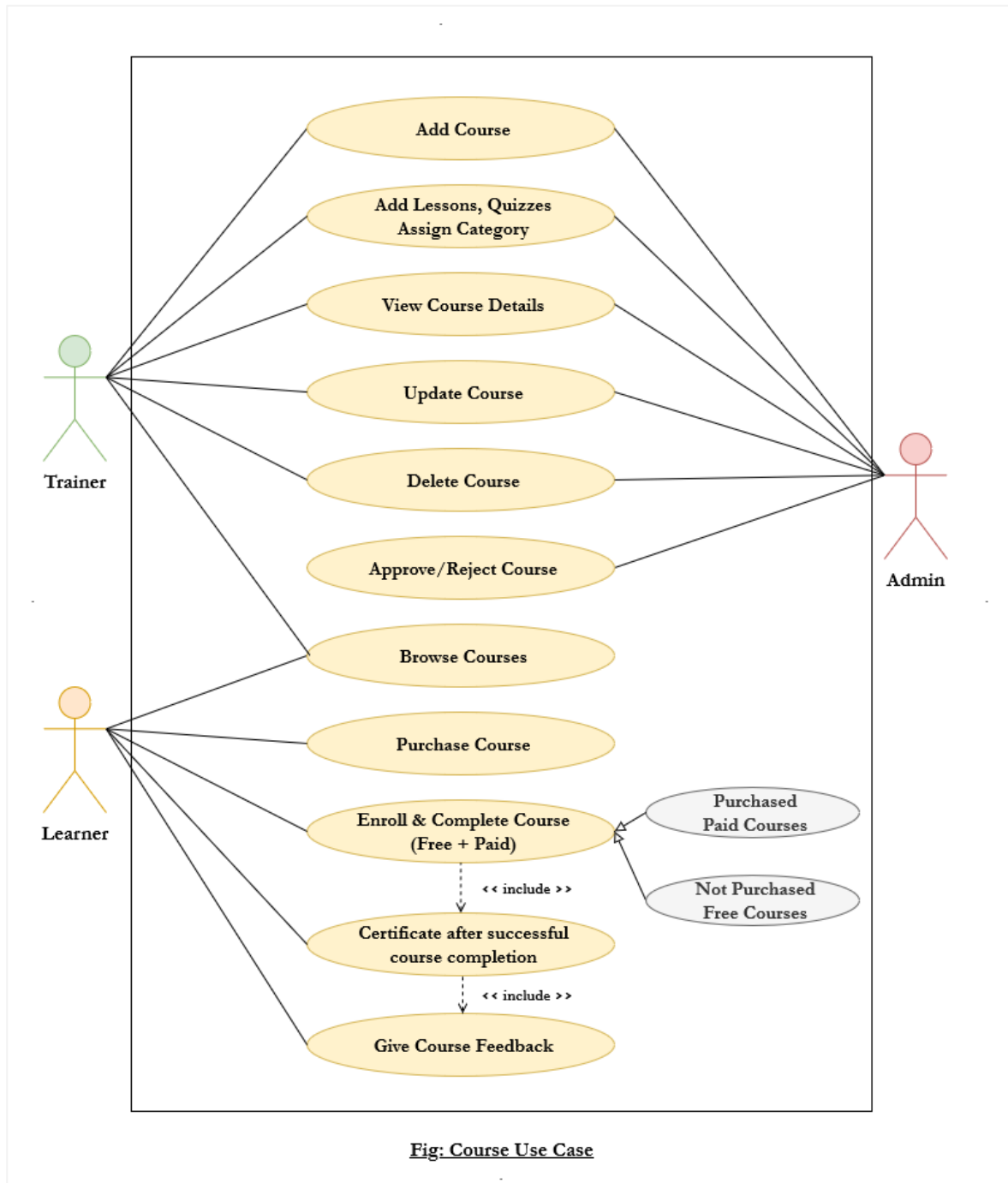
Fig 4.1: Registration Use Case

### 4.1.1 Registration Use Case Description

Table 4.1.1: Registration Use case Description

Use Case ID	UC-01
Use Case Name	Registration
Scenario	User want to register or create a new account
Trigger Event	Need to click on register or sign up button
Brief Description	To create an account users need to fill-up some required information like: first name, last name, email, dob, phone, password and agree with terms & conditions.
Actors	Learner, Trainer, Customer
Extending Use Case	None
Included Use Case	None
Stakeholders	Learner, Trainer, Customer
Pre-Condition	User must need an email address to register in the system
Basic Flow	<ul style="list-style-type: none"> <li>✓ Click on register or signup button</li> <li>✓ Enter into the page</li> <li>✓ Fill-up all the required fields and agree with terms</li> <li>✓ Click on create account</li> <li>✓ Confirm email from user mail account</li> <li>✓ Registered successfully</li> </ul>
Alternative Path	None
Exceptional Path	None
Post-Condition	All the required field must be validate

## 4.2 Course Implementation Use Case Diagram



**NOTE:** For better view please follow the link – [Course Use Case](#)

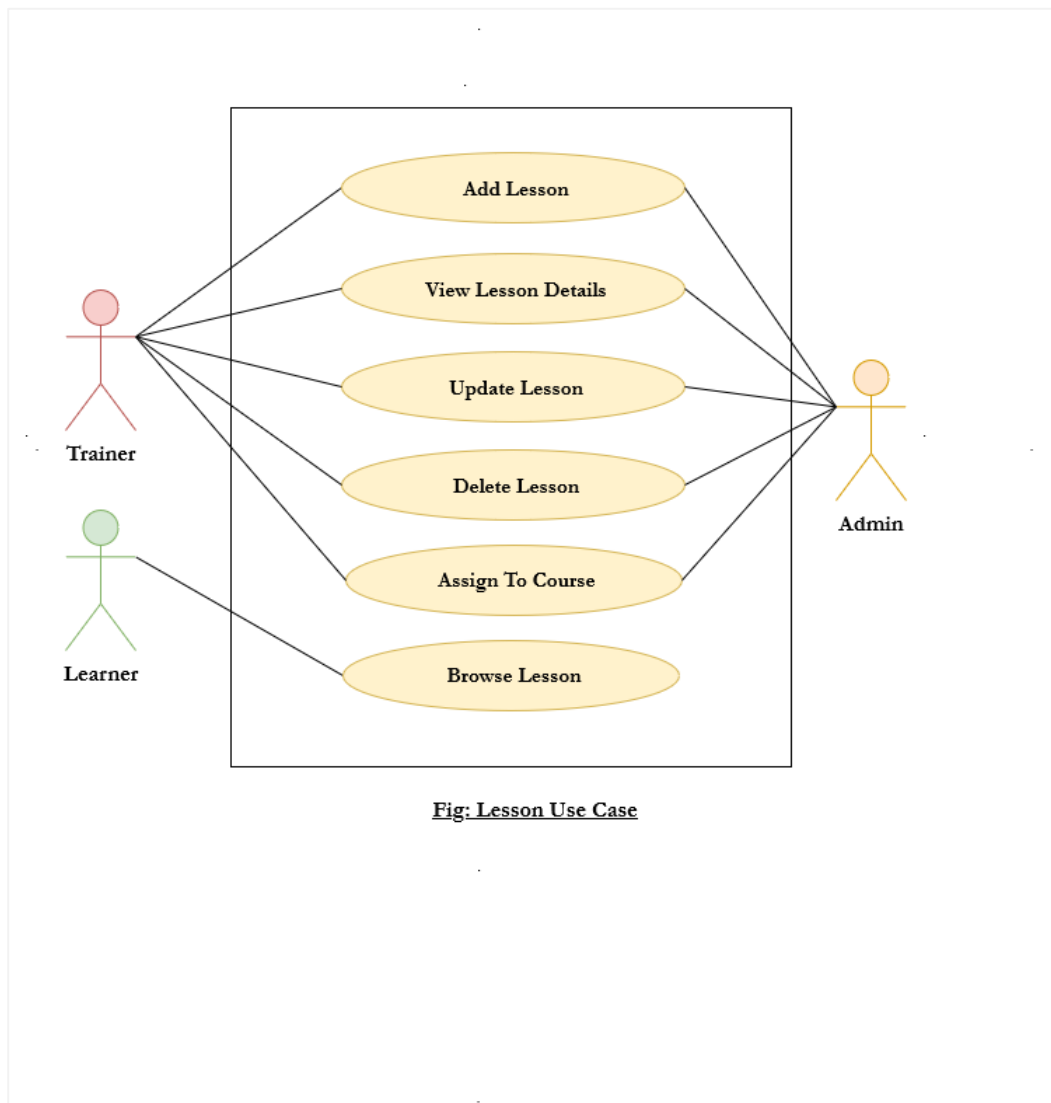
Fig 4.2: Course Implementation Use Case

#### 4.2.1 Course Implementation Use Case Description

Table 4.2.1: Course Implementation Use Case Description

Use Case ID	UC-02				
Use Case Name	Course Implementation				
Scenario	Trainer or Admin want to do course related functionality. Learners want to browse and execute course related staffs				
Trigger Event	Need to click on course related modules				
Brief Description	To do course related stuff they need to browse course modules and fill-up all the required fields to add/update course, learners need to enroll to complete course and after complete course will get certificate and can give feedback on course.				
Actors	Learner, Trainer, Admin				
Extending Use Case	None				
Included Use Case	Login Use Case				
Stakeholders	Learner, Trainer, Admin				
Pre-Condition	Users must authenticate with valid credentials and should have the related privileges to do the required stuff.				
Basic Flow	<table border="1"> <tr> <td>Trainers and Admin</td> <td> <ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul> </td> </tr> <tr> <td>Learners</td> <td> <ul style="list-style-type: none"> <li>✓ Browse courses</li> <li>✓ Enroll paid or free courses</li> <li>✓ Complete courses</li> <li>✓ Get certificate</li> <li>✓ Give feedback on course</li> </ul> </td> </tr> </table>	Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul>	Learners	<ul style="list-style-type: none"> <li>✓ Browse courses</li> <li>✓ Enroll paid or free courses</li> <li>✓ Complete courses</li> <li>✓ Get certificate</li> <li>✓ Give feedback on course</li> </ul>
Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul>				
Learners	<ul style="list-style-type: none"> <li>✓ Browse courses</li> <li>✓ Enroll paid or free courses</li> <li>✓ Complete courses</li> <li>✓ Get certificate</li> <li>✓ Give feedback on course</li> </ul>				
Alternative Path	None				
Exceptional Path	None				
Post-Condition	All the required fields must be validated. Learners need to enroll and complete the course to get a certificate.				

### 4.3 Lesson Implementation Use Case Diagram



**NOTE:** For better view please follow the link – [Lesson Use Case](#)

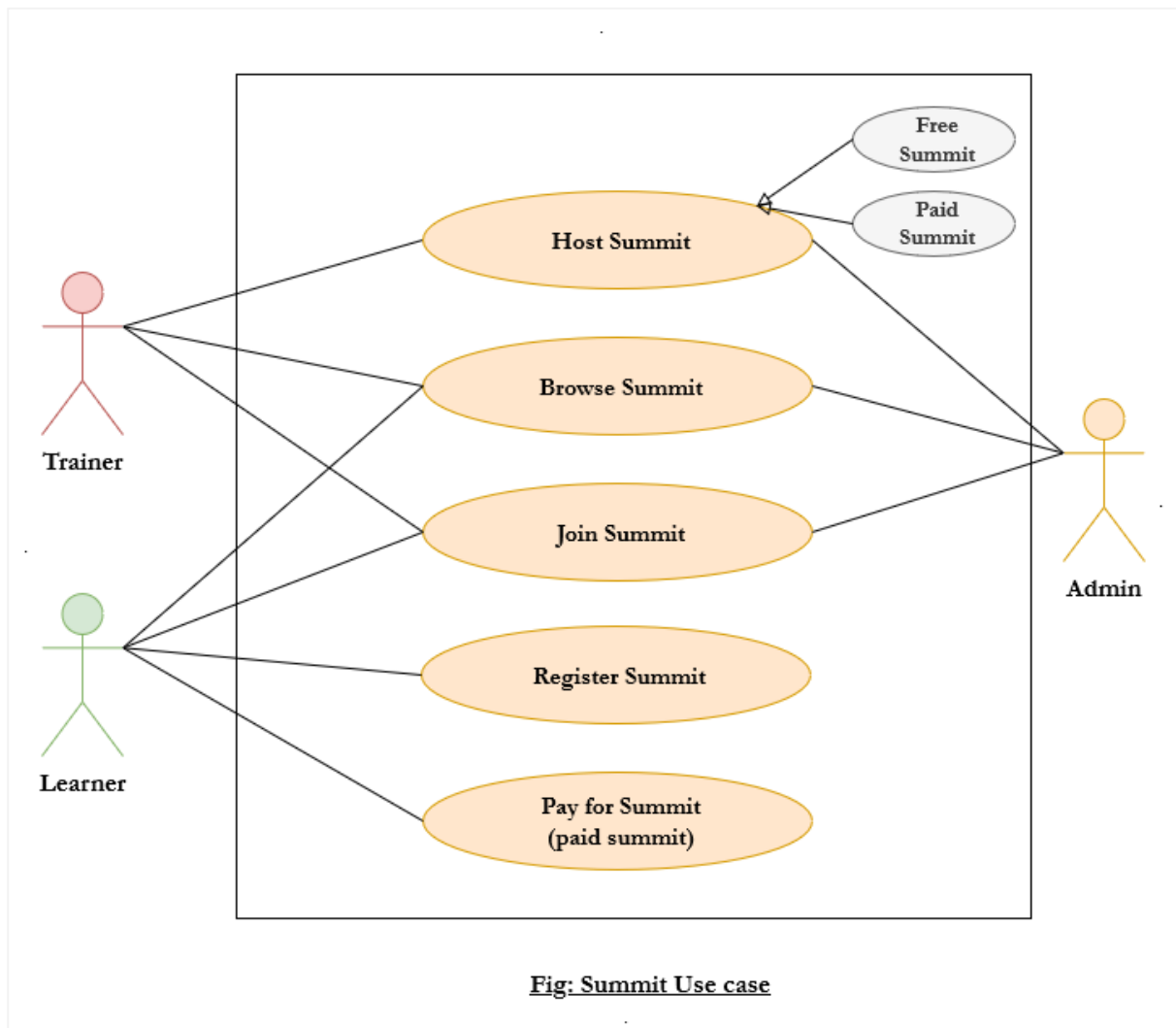
Fig 4.3: Lesson Use Case

### 4.3.1 Lesson Implementation Use Case Descriptions

Table 4.3.1: Lesson Implementation Use Case Descriptions

Use Case ID	UC-03				
Use Case Name	Lesson Implementation				
Scenario	Trainer or Admin want to do lessons related to functionality. Learners want to browse lessons.				
Trigger Event	Need to click on course related modules				
Brief Description	To do lessons related to stuff, admin and trainer need to enter lesson modules and fill-up all the required fields to add/update lessons, learners need to enroll into a course to browse the related lessons.				
Actors	Learner, Trainer, Admin				
Extending Use Case	None				
Included Use Case	Login Use Case				
Stakeholders	Learner, Trainer, Admin				
Pre-Condition	Users must authenticate with valid credentials and should have the related privileges to do the required stuff.				
Basic Flow	<table border="1"> <tr> <td>Trainers and Admin</td> <td> <ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul> </td> </tr> <tr> <td>Learners</td> <td> <ul style="list-style-type: none"> <li>✓ Enroll a course</li> <li>✓ Browse course related lessons</li> <li>✓ Complete lessons</li> </ul> </td> </tr> </table>	Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul>	Learners	<ul style="list-style-type: none"> <li>✓ Enroll a course</li> <li>✓ Browse course related lessons</li> <li>✓ Complete lessons</li> </ul>
Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete course module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields or execute related functionality</li> </ul>				
Learners	<ul style="list-style-type: none"> <li>✓ Enroll a course</li> <li>✓ Browse course related lessons</li> <li>✓ Complete lessons</li> </ul>				
Alternative Path	None				
Exceptional Path	None				
Post-Condition	All the required fields must be validated. Learners need to enroll in a course to browse related lessons.				

#### 4.4 Summit Use Case Diagram



**NOTE:** For better view please follow the link – [Summit Use Case](#)

Fig 4.4: Summit Use Case

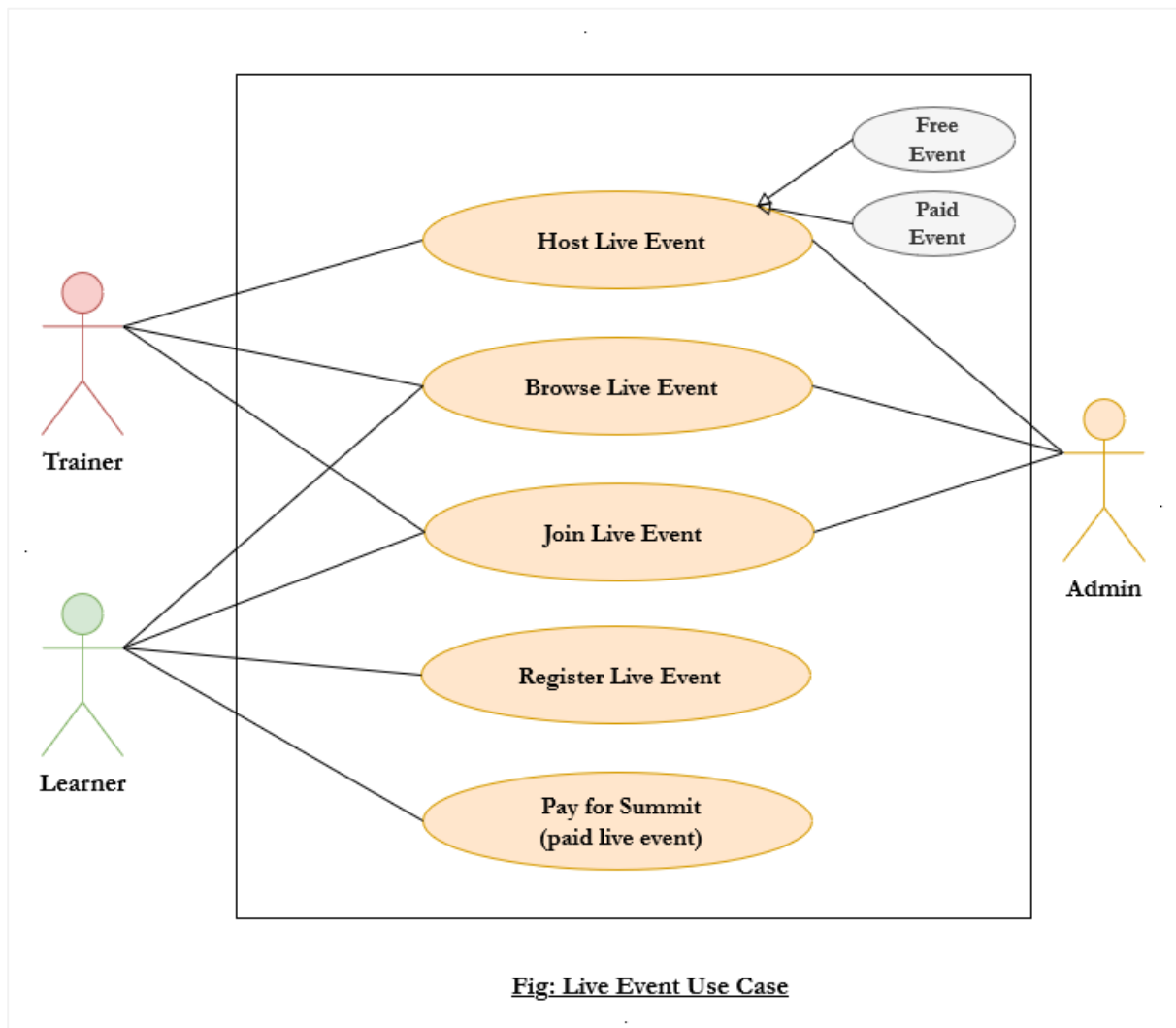


#### 4.4.1 Summit Use Case Description

Table: 4.4.1: Summit Use Case Description

Use Case ID	UC-04				
Use Case Name	Summit Implementation				
Scenario	Trainer or Admin want to do summit related functionality. Learners want to join the summit.				
Trigger Event	Need to click on summit related modules				
Brief Description	To do summit related stuff admin and trainer need to enter on summit modules and fill-up all the required fields to add/update summit, to join summit learners need to register or can join without registering.				
Actors	Learner, Trainer, Admin				
Extending Use Case	None				
Included Use Case	None				
Stakeholders	Learner, Trainer, Admin				
Pre-Condition	Use must need a valid email for summit registration				
Basic Flow	<table border="1"> <tr> <td>Trainers and Admin</td> <td> <ul style="list-style-type: none"> <li>✓ Click on add/update/delete summit module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul> </td> </tr> <tr> <td>Learners</td> <td> <ul style="list-style-type: none"> <li>✓ Register for summit (if required)</li> <li>✓ Join summit</li> </ul> </td> </tr> </table>	Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete summit module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>	Learners	<ul style="list-style-type: none"> <li>✓ Register for summit (if required)</li> <li>✓ Join summit</li> </ul>
Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete summit module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>				
Learners	<ul style="list-style-type: none"> <li>✓ Register for summit (if required)</li> <li>✓ Join summit</li> </ul>				
Alternative Path	None				
Exceptional Path	None				
Post-Condition	All the required fields must be validated.				

## 4.5 Live Event Use Case Diagram



**NOTE:** For better view please follow the link – [Live Event Use Case](#)

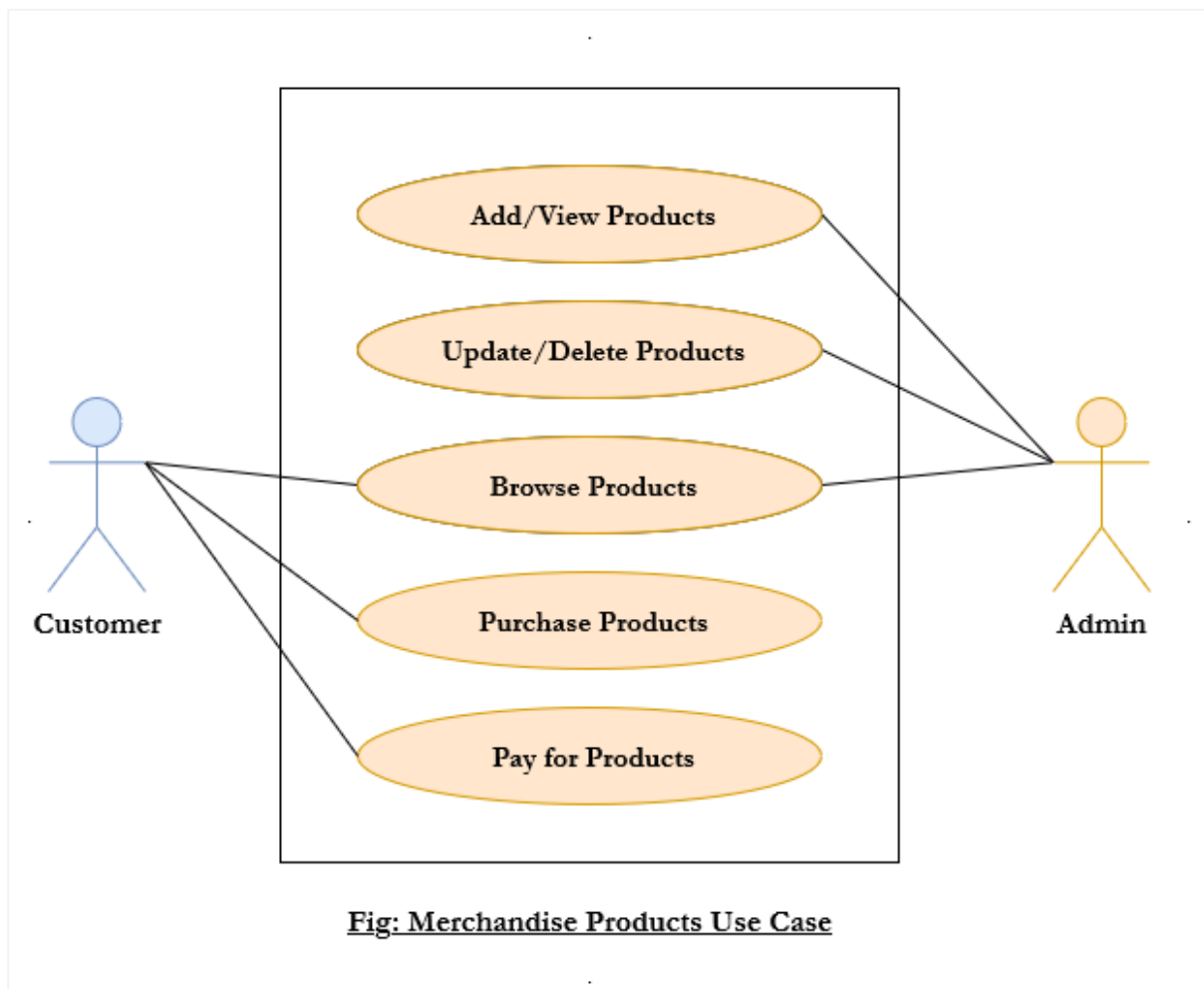
Fig 4.5: Live Event use Case

#### 4.5.1 Live Event Use Case Description

Table 4.5.1: Live Event Use Case Description

Use Case ID	UC-05				
Use Case Name	Live Event Implementation				
Scenario	Trainer or Admin want to do live event related functionality. Learners want to join live events.				
Trigger Event	Need to click on live event related modules				
Brief Description	To do live event related stuff admin and trainer need to enter on live event modules and fill-up all the required fields to add/update live events, to join live event learners need to register or can join without registering.				
Actors	Learner, Trainer, Admin				
Extending Use Case	None				
Included Use Case	None				
Stakeholders	Learner, Trainer, Admin				
Pre-Condition	Use must need a valid email for live event registration				
Basic Flow	<table border="1"> <tr> <td>Trainers and Admin</td> <td> <ul style="list-style-type: none"> <li>✓ Click on add/update/delete live event module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul> </td> </tr> <tr> <td>Learners</td> <td> <ul style="list-style-type: none"> <li>✓ Register for live event (if required)</li> <li>✓ Join Live Event</li> </ul> </td> </tr> </table>	Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete live event module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>	Learners	<ul style="list-style-type: none"> <li>✓ Register for live event (if required)</li> <li>✓ Join Live Event</li> </ul>
Trainers and Admin	<ul style="list-style-type: none"> <li>✓ Click on add/update/delete live event module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>				
Learners	<ul style="list-style-type: none"> <li>✓ Register for live event (if required)</li> <li>✓ Join Live Event</li> </ul>				
Alternative Path	None				
Exceptional Path	None				
Post-Condition	All the required fields must be validated.				

## 4.6 Merchandise Product Use Case Diagram



**NOTE:** For better view please follow the link – [Merchandise Product Use Case](#)

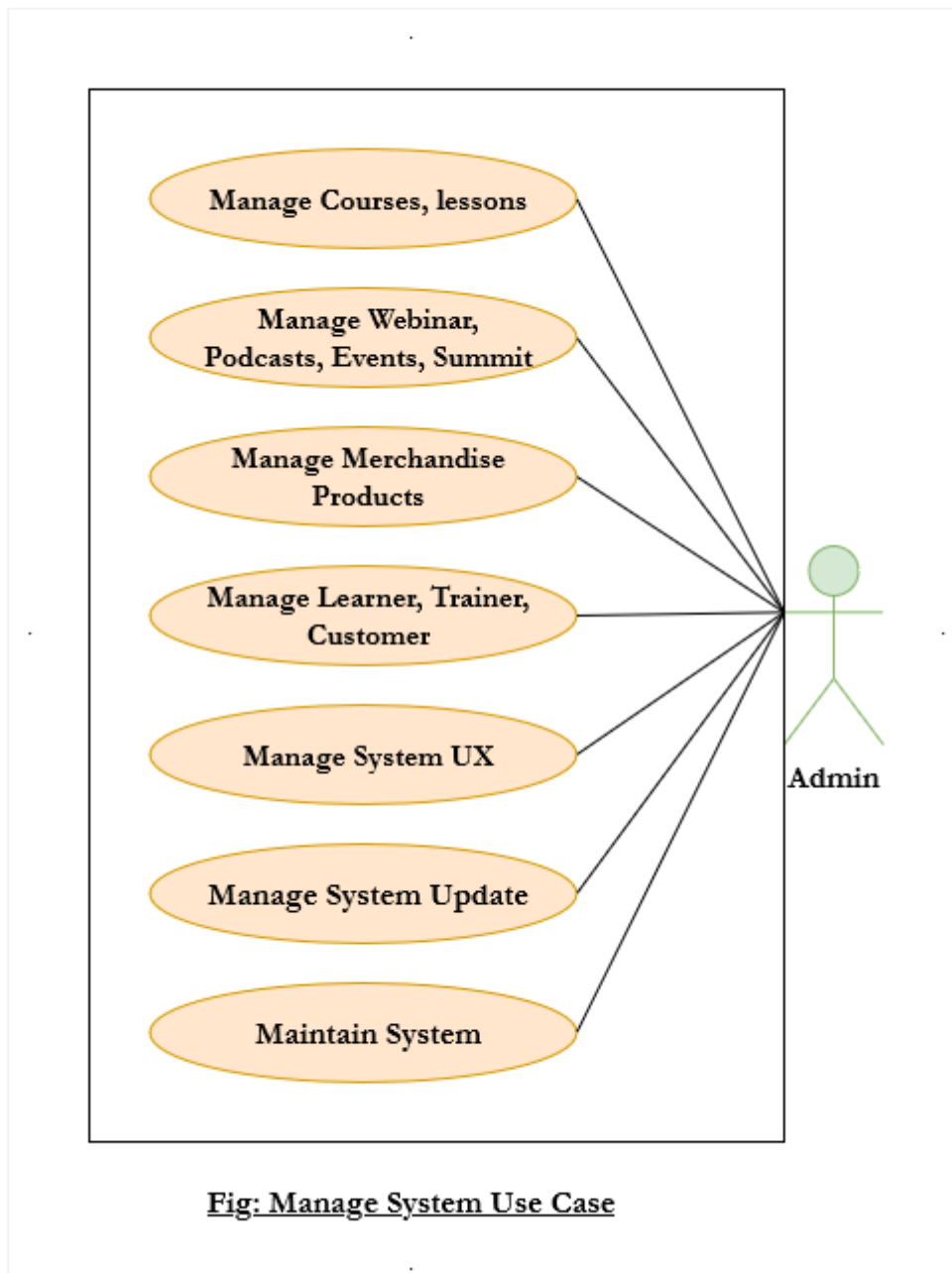
Fig 4.6: Merchandise Products Use Case

#### 4.6.1 Merchandise Products Use Case Description

Table 4.6.1: Merchandise Products Use Case Description

Use Case ID	UC-06					
Use Case Name	Merchandise Products Implementation					
Scenario	Admin wants to create merchandise products. Customers want to buy merchandise products					
Trigger Event	Need to click on merchandise related modules					
Brief Description	To do merchandise related staff admin need to enter on merchandise modules. Fill-up all the required fields to create/update merchandise products, to purchase merchandise products customers need to login in the system.					
Actors	Admin, Customer					
Extending Use Case	None					
Included Use Case	None					
Stakeholders	Admin, Customer					
Pre-Condition	Need to authenticate to purchase products.					
Basic Flow	<table border="1"> <tr> <td>Admin</td> <td> <ul style="list-style-type: none"> <li>✓ Click on create/update/delete merchandise module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul> </td> </tr> <tr> <td>Customers</td> <td> <ul style="list-style-type: none"> <li>✓ Browse merchandise products</li> <li>✓ Add to cart</li> <li>✓ Go to purchase and provide details</li> <li>✓ Pay for products and purchase successfully.</li> </ul> </td> </tr> </table>		Admin	<ul style="list-style-type: none"> <li>✓ Click on create/update/delete merchandise module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>	Customers	<ul style="list-style-type: none"> <li>✓ Browse merchandise products</li> <li>✓ Add to cart</li> <li>✓ Go to purchase and provide details</li> <li>✓ Pay for products and purchase successfully.</li> </ul>
Admin	<ul style="list-style-type: none"> <li>✓ Click on create/update/delete merchandise module</li> <li>✓ Enter into module</li> <li>✓ Fill-up all required fields to create or do related functionality</li> </ul>					
Customers	<ul style="list-style-type: none"> <li>✓ Browse merchandise products</li> <li>✓ Add to cart</li> <li>✓ Go to purchase and provide details</li> <li>✓ Pay for products and purchase successfully.</li> </ul>					
Alternative Path	None					
Exceptional Path	None					
Post-Condition	All the required fields must be validated in time to create/update merchandise products or provide product purchase details.					

## 4.7 Manage System Use Case Diagram



**NOTE:** For better view please follow the link – [Manage System Use Case](#)

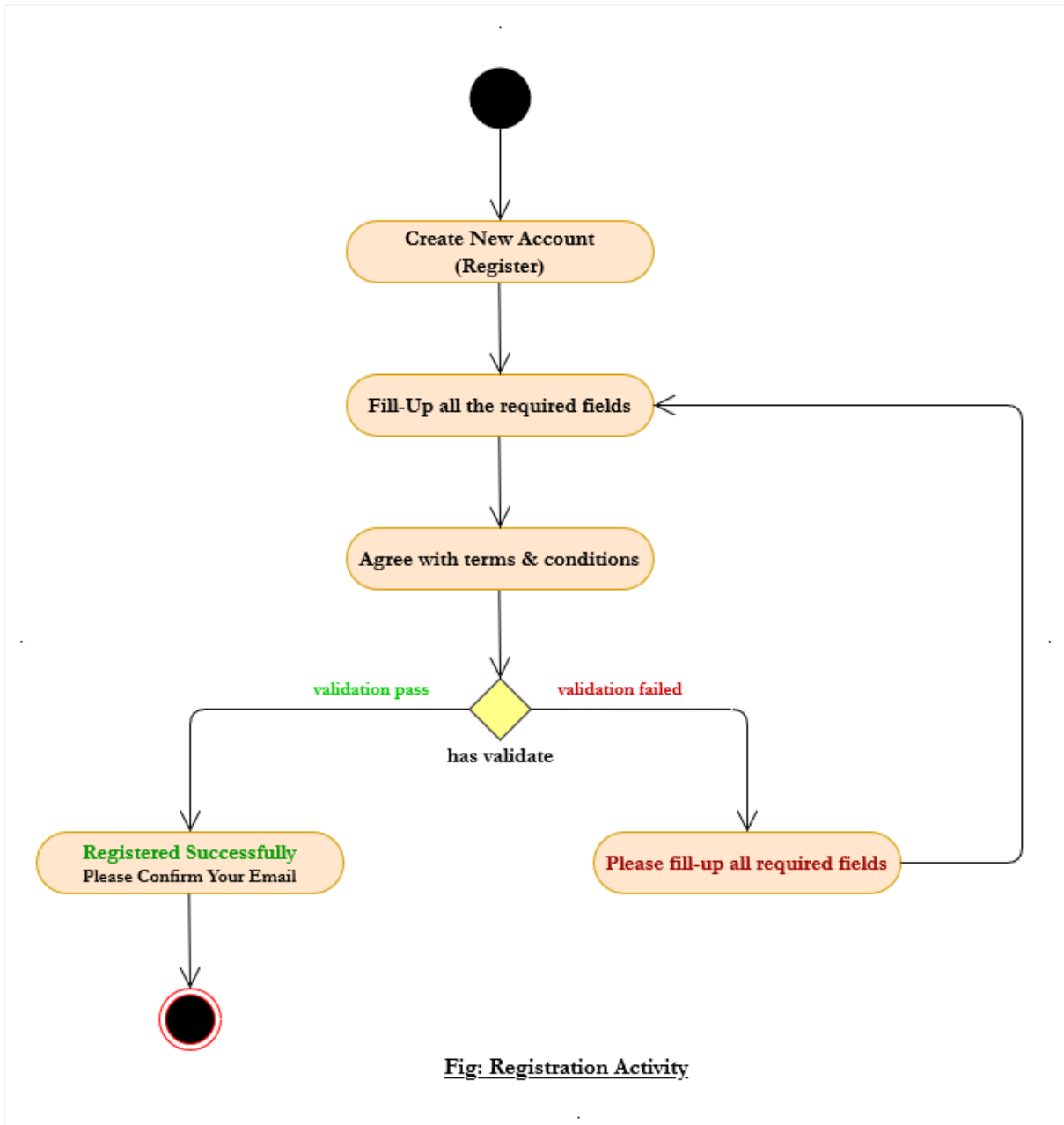
Fig: 4.7 Manage System Use Case

#### 4.7.1 Manage System Use Case Description

Table 4.7.1: Manage System Use Case Description

Use Case ID	UC-07
Use Case Name	Manage System
Scenario	Admin want to manage any module of the system
Trigger Event	Need to click on required module option
Brief Description	To manage any module of the system admin needs to enter into the required module and can do related staffs to the related module.
Actors	Admin
Extending Use Case	None
Included Use Case	None
Stakeholders	Admin
Pre-Condition	Admin must authenticate to manage all the modules
Basic Flow	<ul style="list-style-type: none"><li>✓ Click on required module</li><li>✓ Enter into the module</li><li>✓ Can do the module related staffs</li></ul>
Alternative Path	None
Exceptional Path	None
Post-Condition	None

5.1 Registration Activity Diagram

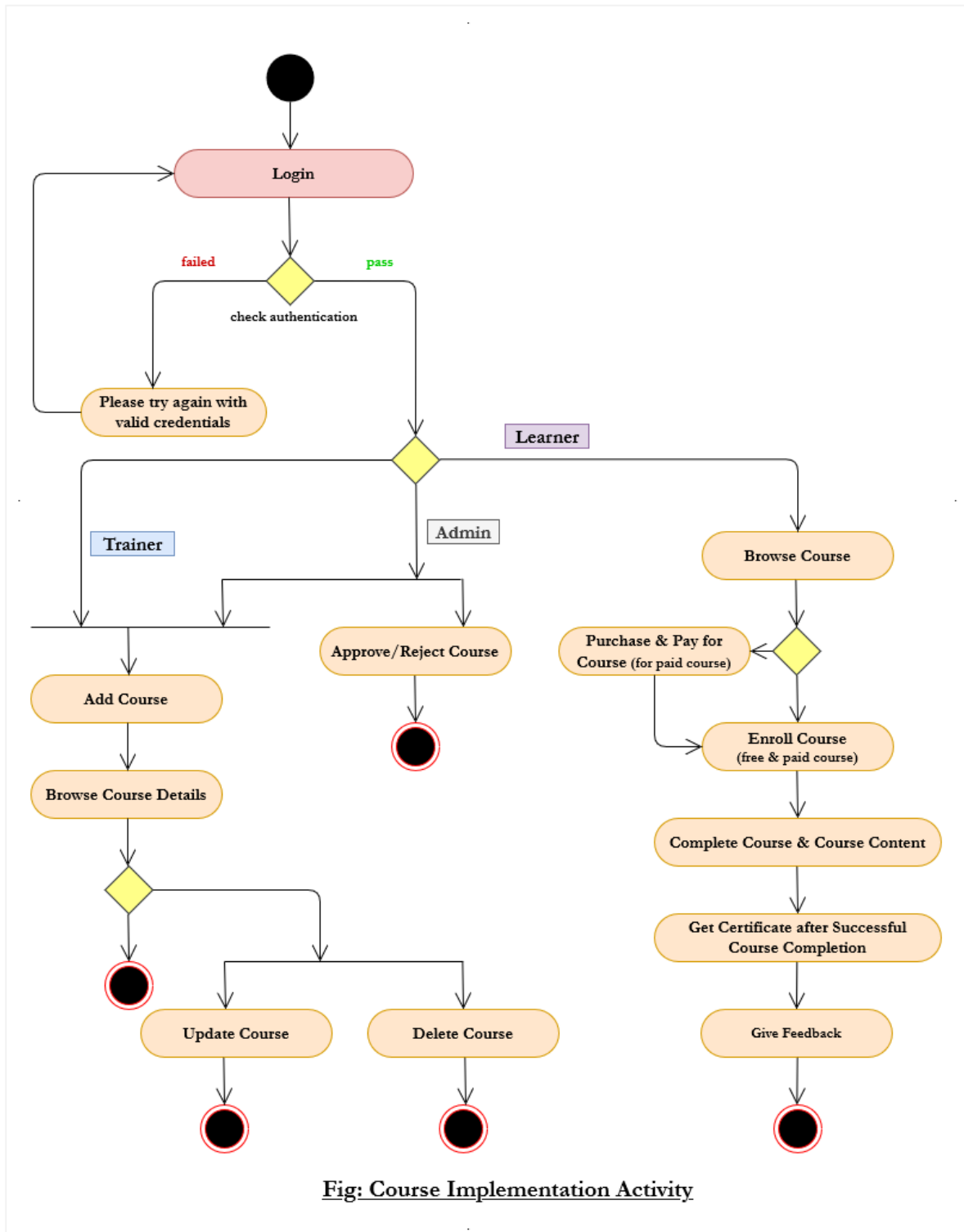


**NOTE:** For better view please follow the link – [Registration Activity](#)

Fig 5.1: Registration Activity Diagram



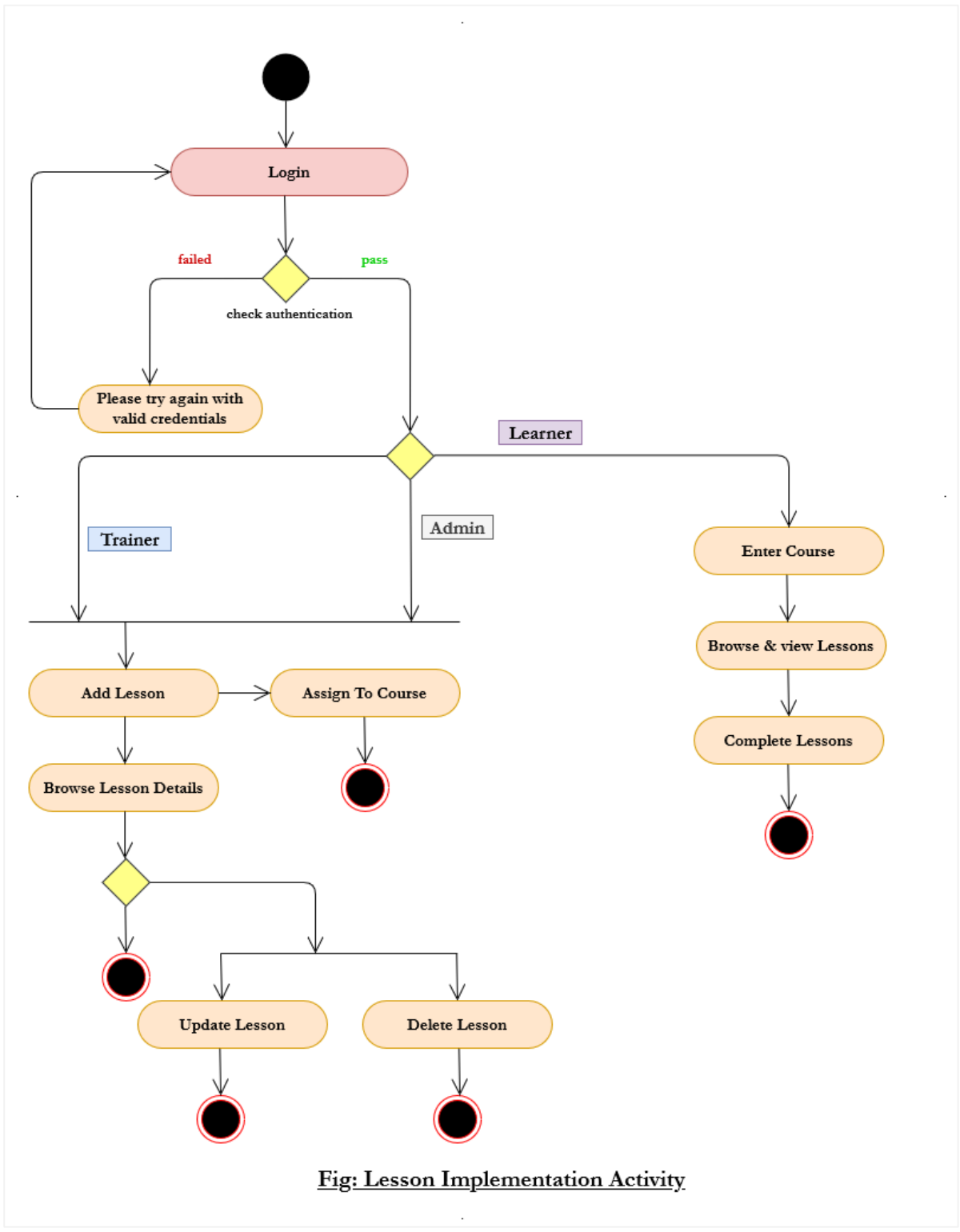
## 5.2 Course Implementation Activity Diagram



**NOTE:** For better view please follow the link – [Course Activity](#)

Fig 5.2: Course Implementation Diagram

### 5.3 Lesson Implementation Activity Diagram

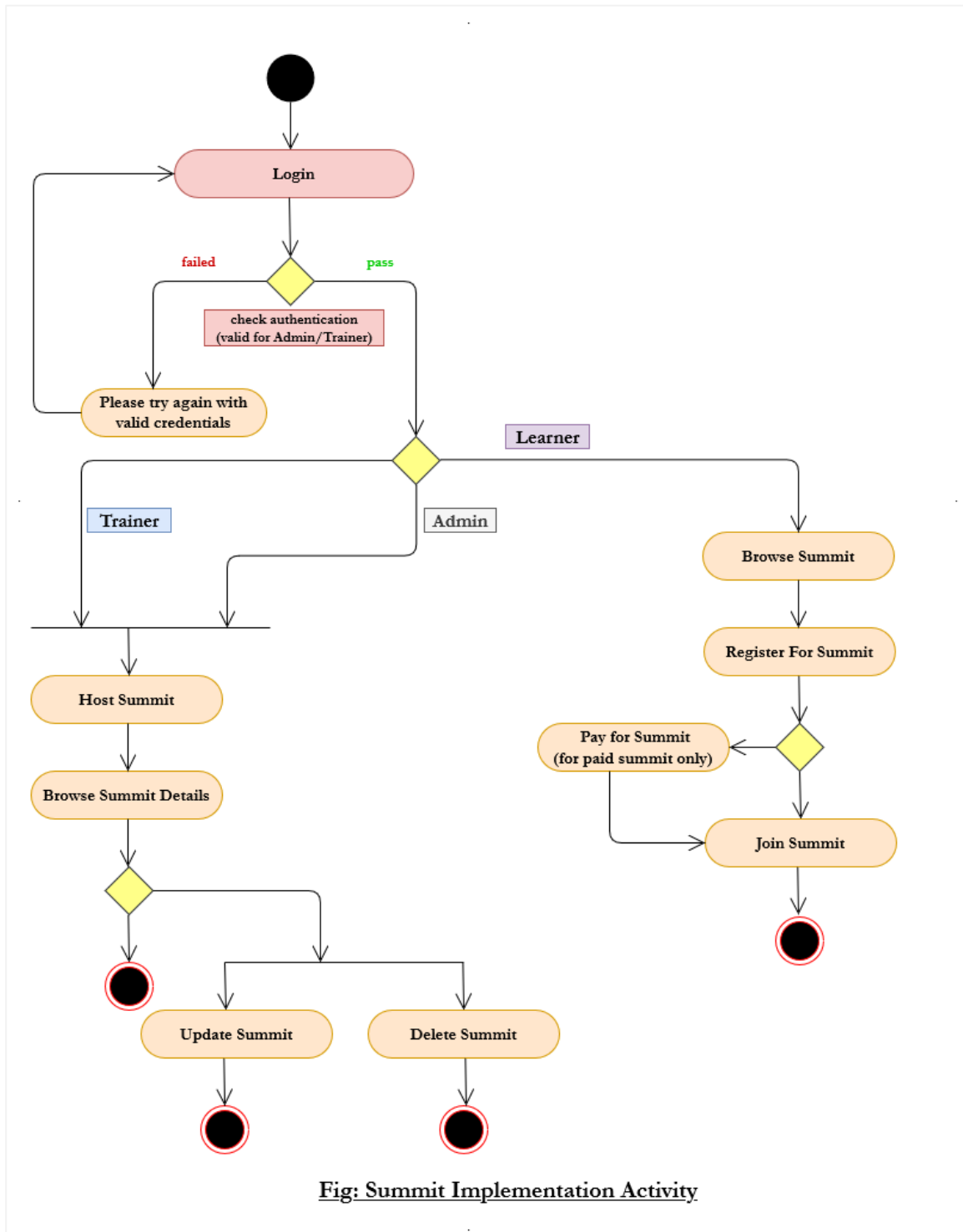


**Fig: Lesson Implementation Activity**

**NOTE:** For better view please follow the link – [Lesson Activity](#)

Fig 5.3: Lesson Implementation Activity Diagram

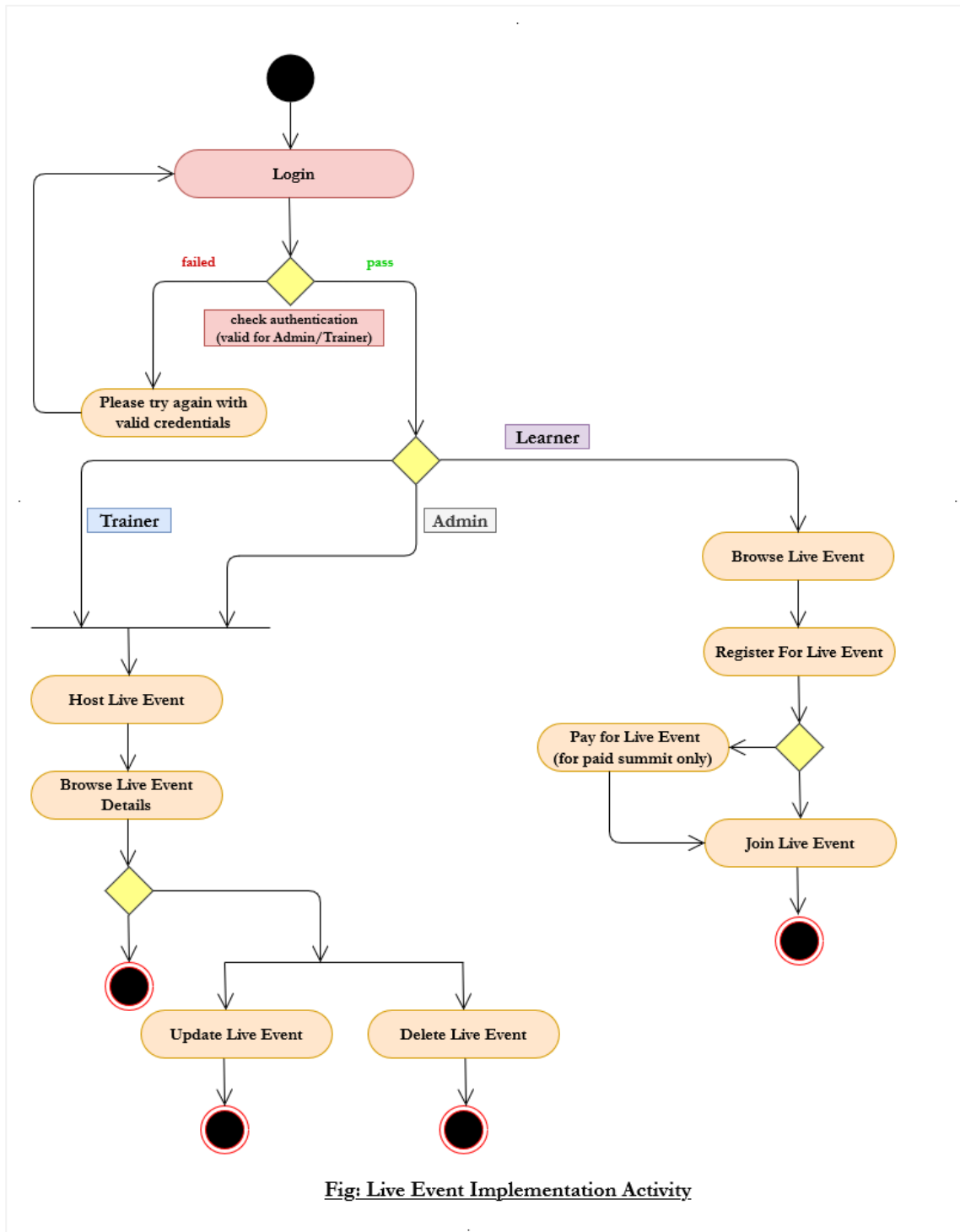
## 5.4 Summit Implementation Activity



**NOTE:** For better view please follow the link – [Summit Activity](#)

Fig: 5.4 Summit Implementation Activity

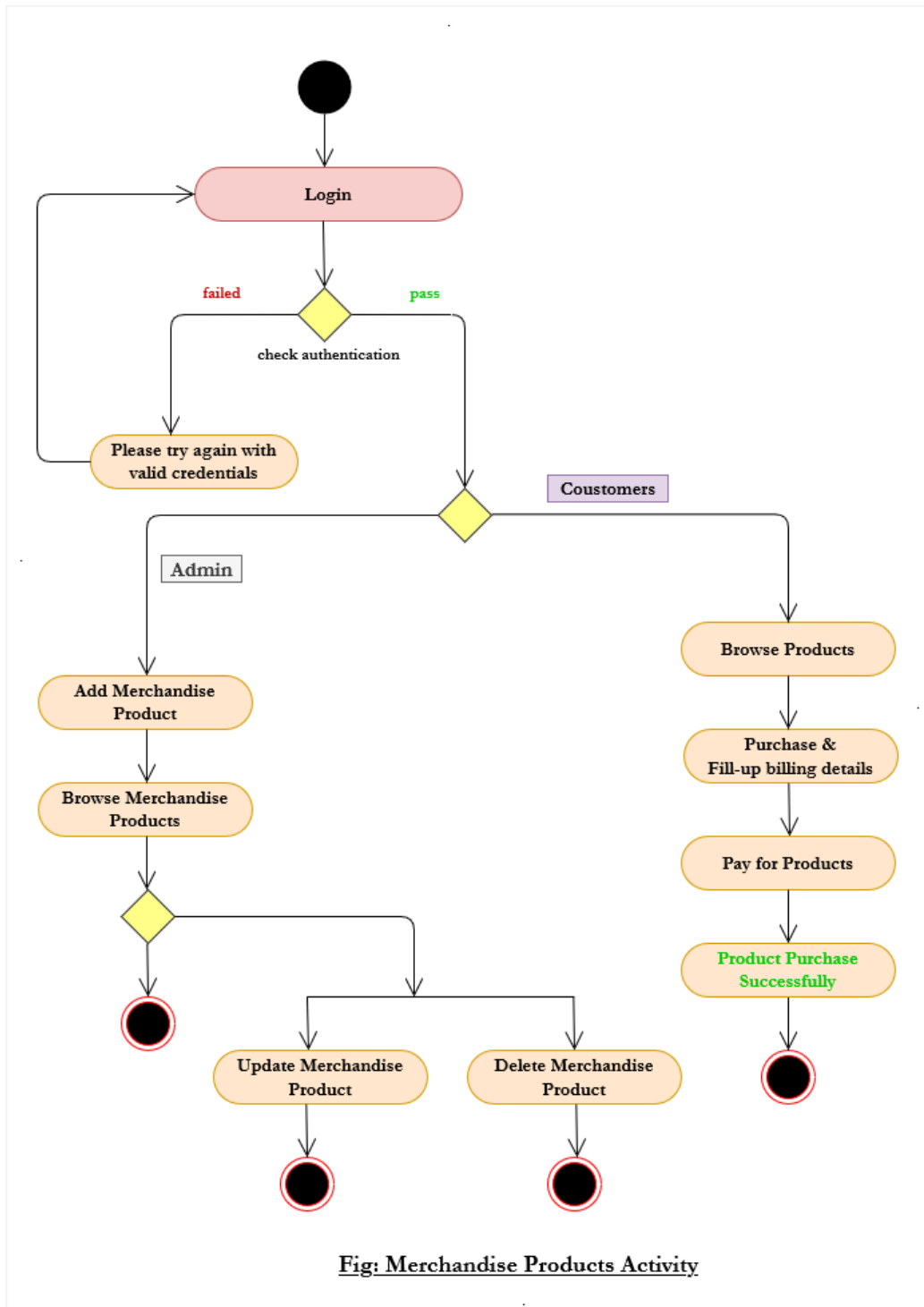
## 5.5 Live Event Implementation Activity



**NOTE:** For better view please follow the link – [Live Event Activity](#)

Fig 5.5: Live Event Implementation Activity

## 5.6 Merchandise Products Activity Diagram

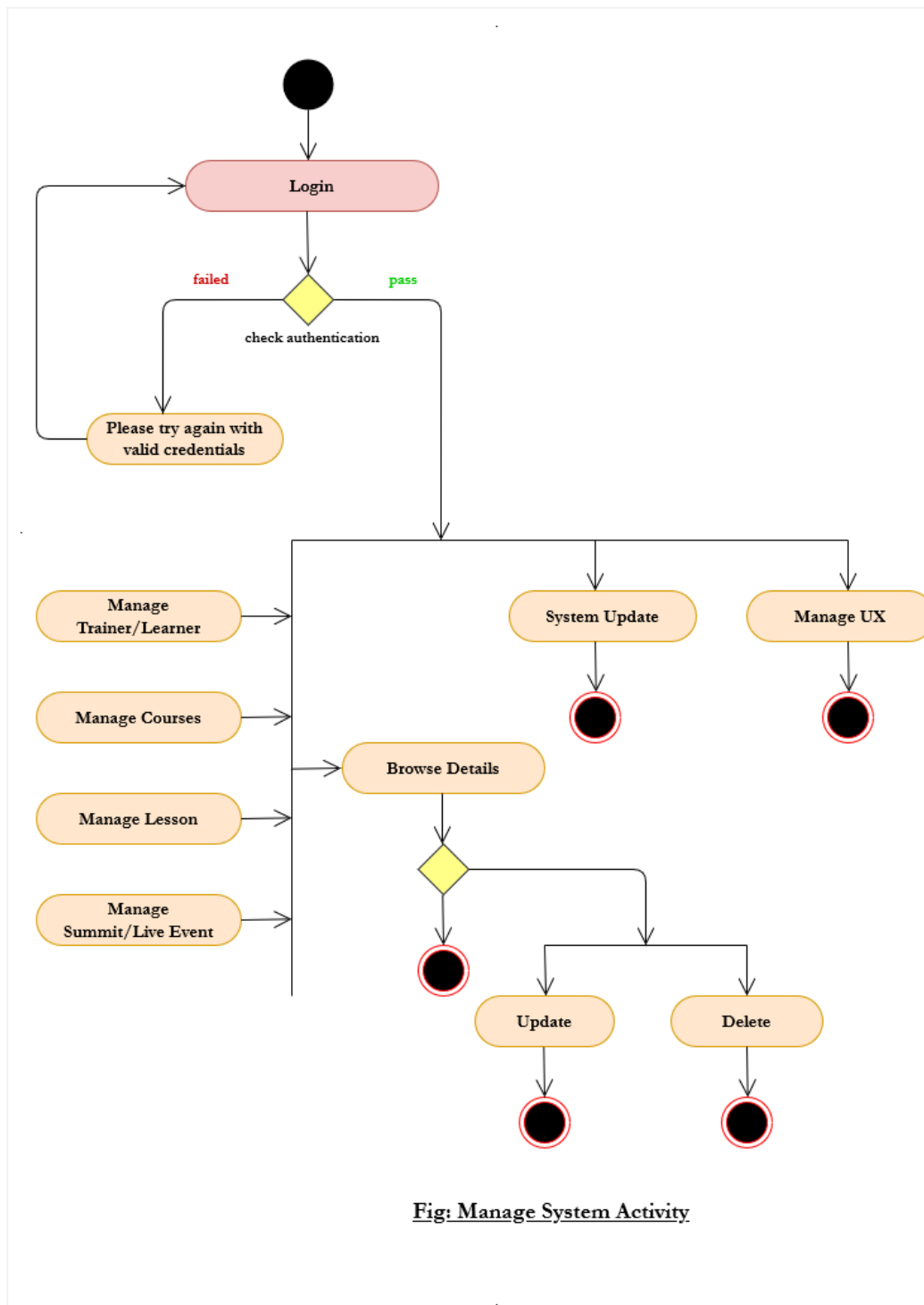


**Fig: Merchandise Products Activity**

**NOTE:** For better view please follow the link – [Merchandise Product Activity](#)

Fig 5.6: Merchandise Product Activity

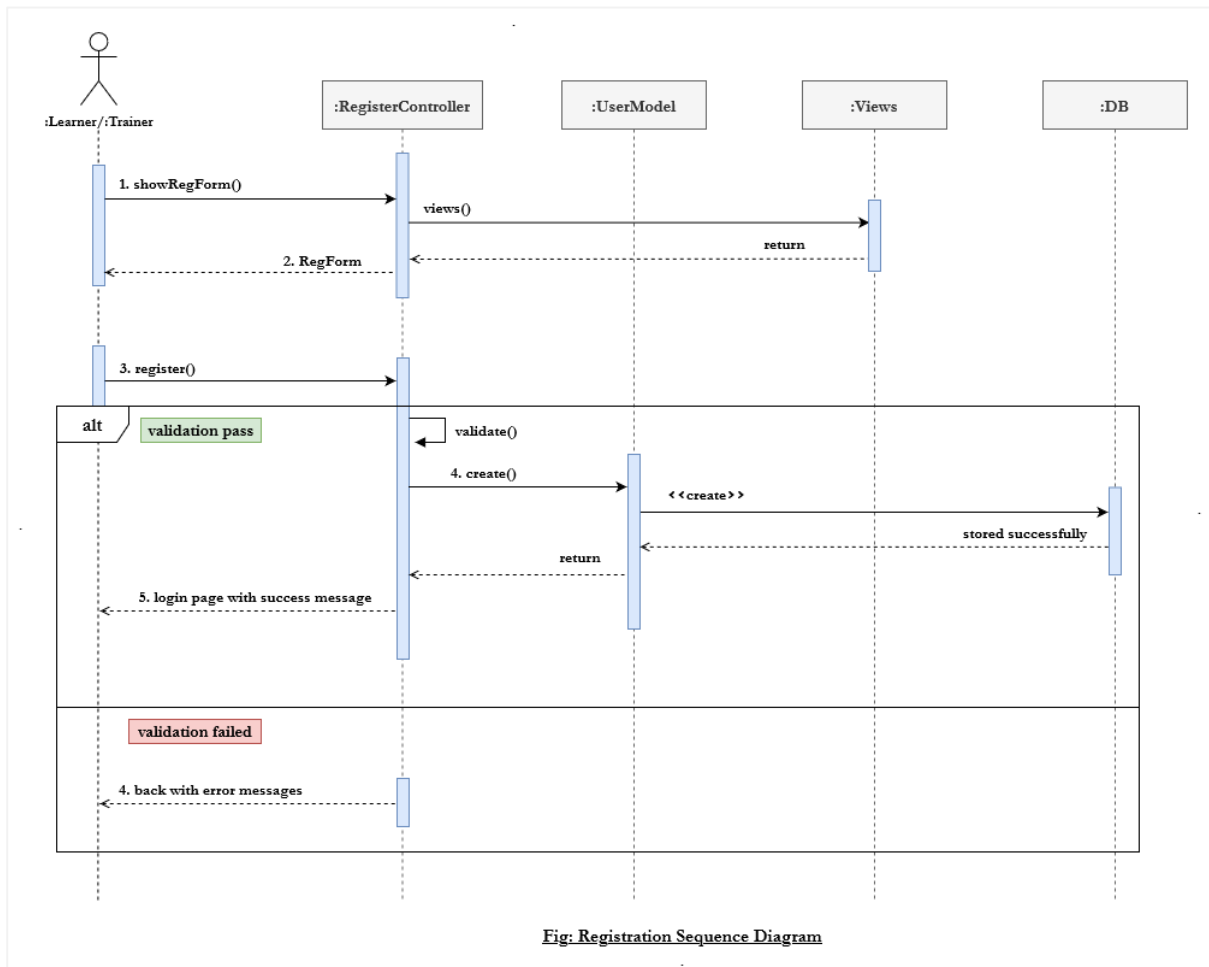
## 5.7 Manage System Activity



**NOTE:** For better view please follow the link – [Manage System Activity](#)

Fig 5.7: Manage System Activity

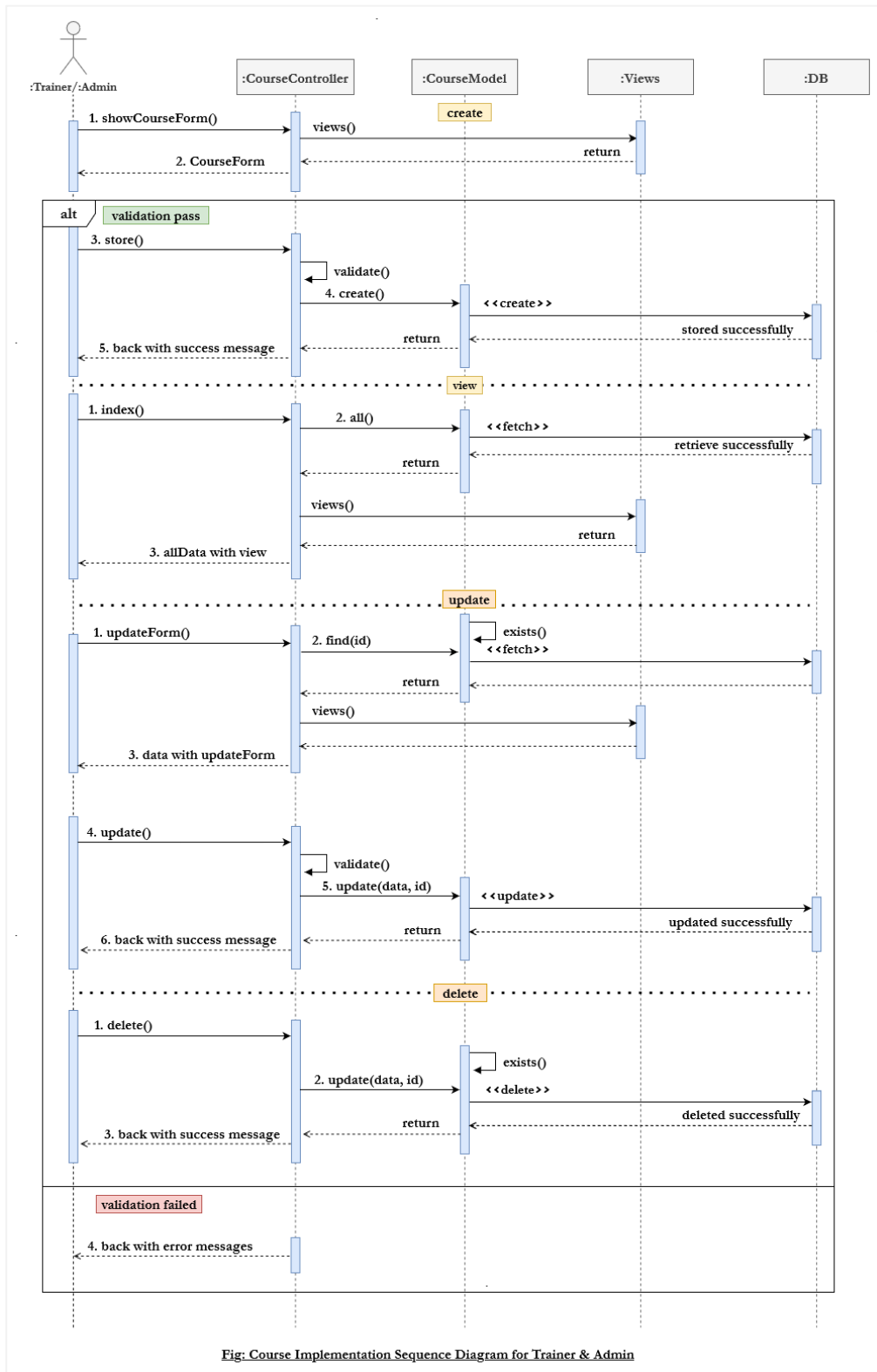
6.1 Registration Sequence Diagram



**NOTE:** For better view please follow the link – [Registration Sequence Diagram](#)

Fig 6.1: Registration Sequence Diagram

## 6.2 Course Implementation Sequence Diagram for Trainer & Admin

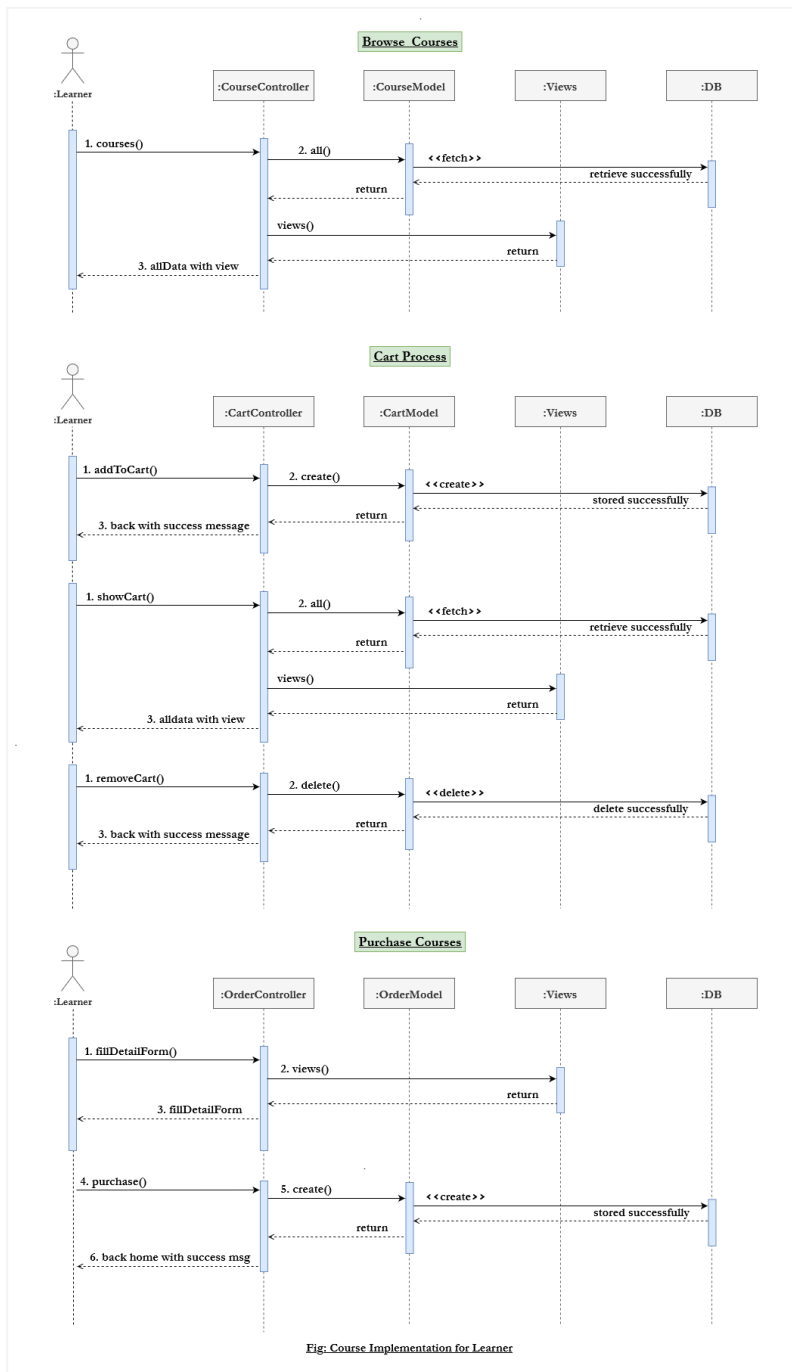


**NOTE:** For better view please follow the link – [Course Sequence Diagram](#)

Fig 6.2: Course Sequence Diagram



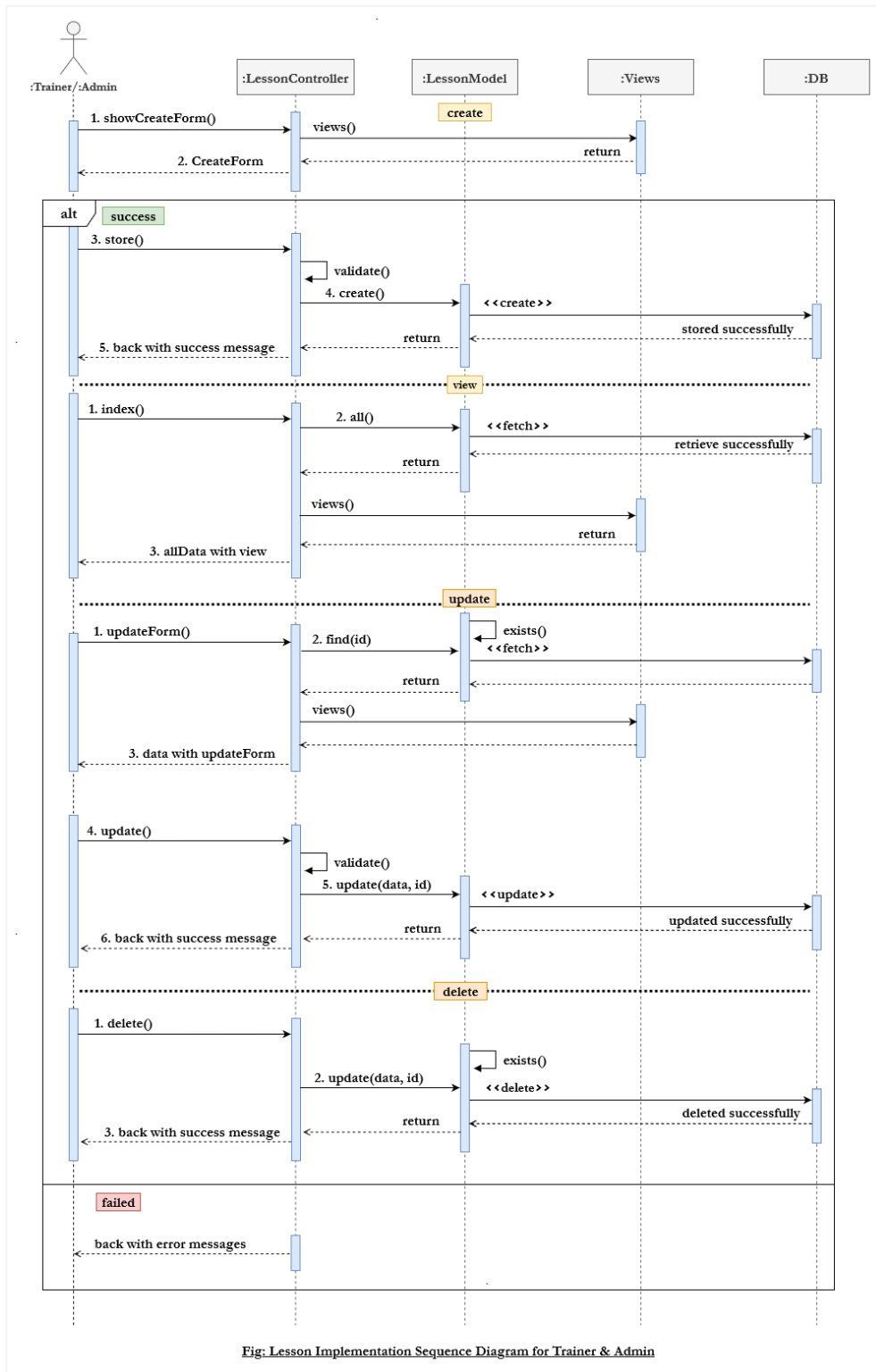
### 6.3 Course Implementation Sequence Diagram for Learner



**NOTE:** For better view please follow the link – [Course Sequence Diagram for Learner](#)

Fig 6.3: Course Implementation Sequence Diagram

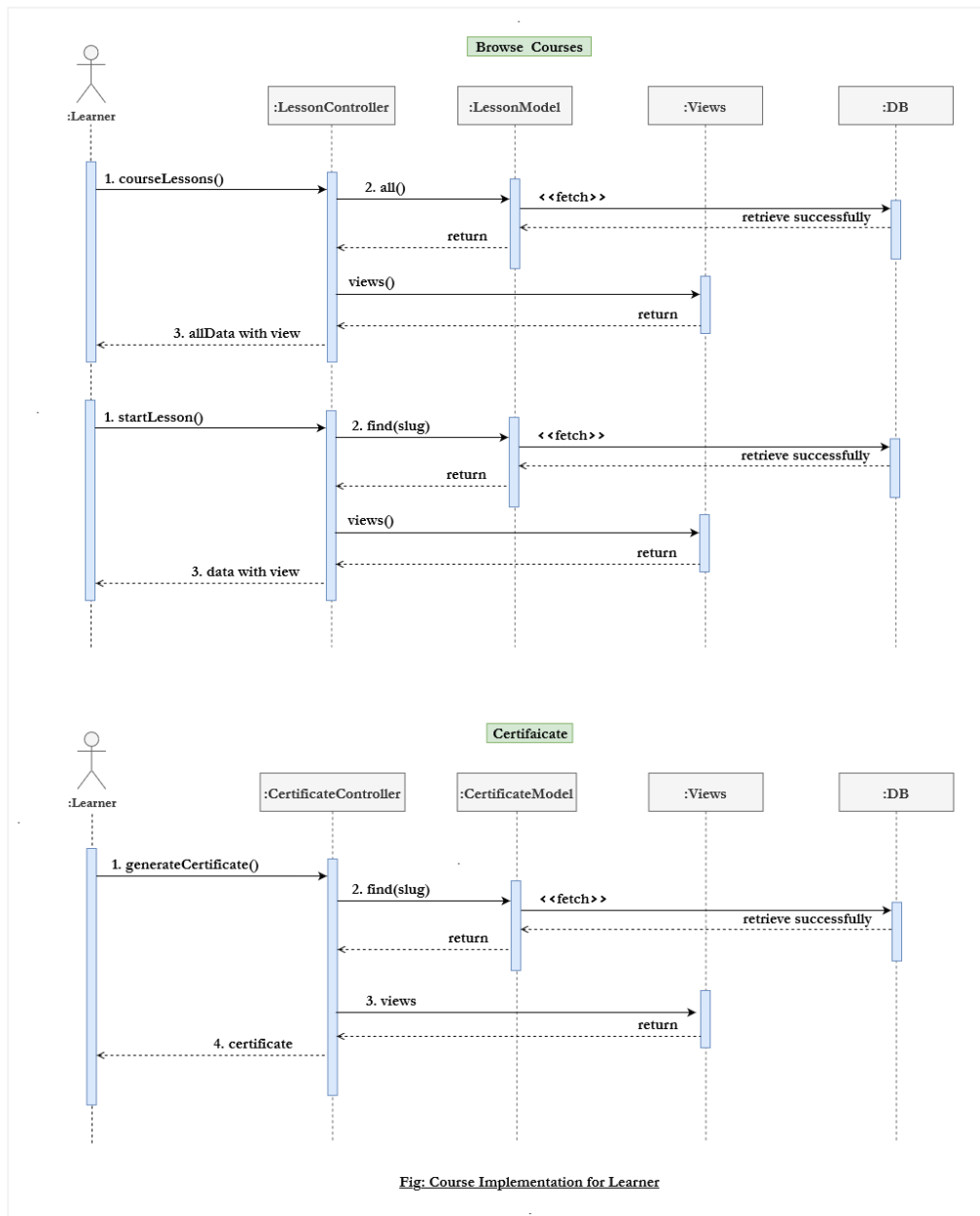
## 6.4 Lesson Implementation Sequence Diagram for Trainer & Admin



**NOTE:** For better view please follow the link – [Lesson Sequence Diagram](#)

Fig 6.4: Lesson Implementation Sequence Diagram

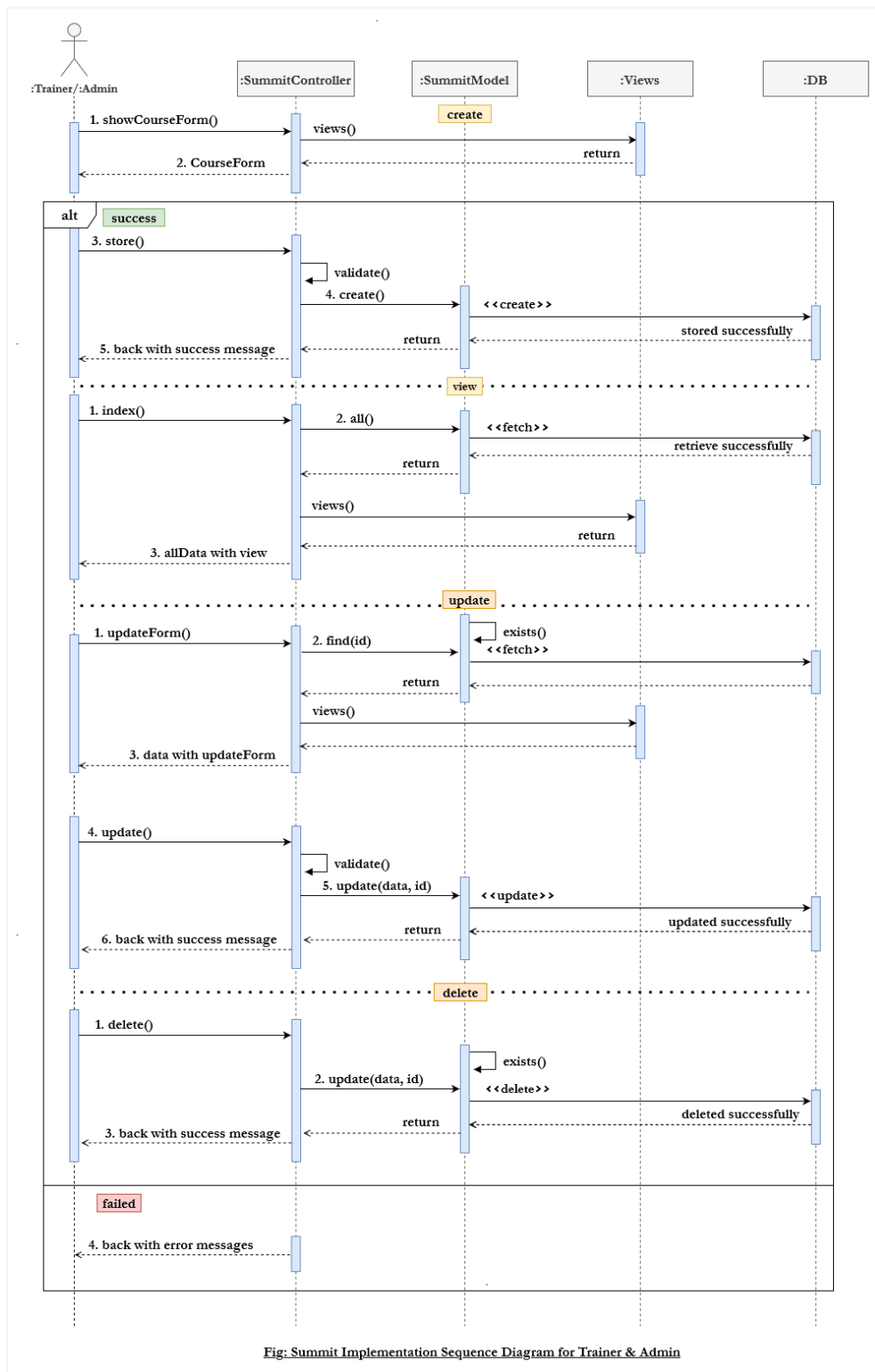
## 6.5 Lesson Implementation Sequence Diagram for Learner



**NOTE:** For better view please follow the link – [Lesson Sequence Diagram for Learner](#)

Fig 6.5: Lesson Implementation Sequence Diagram

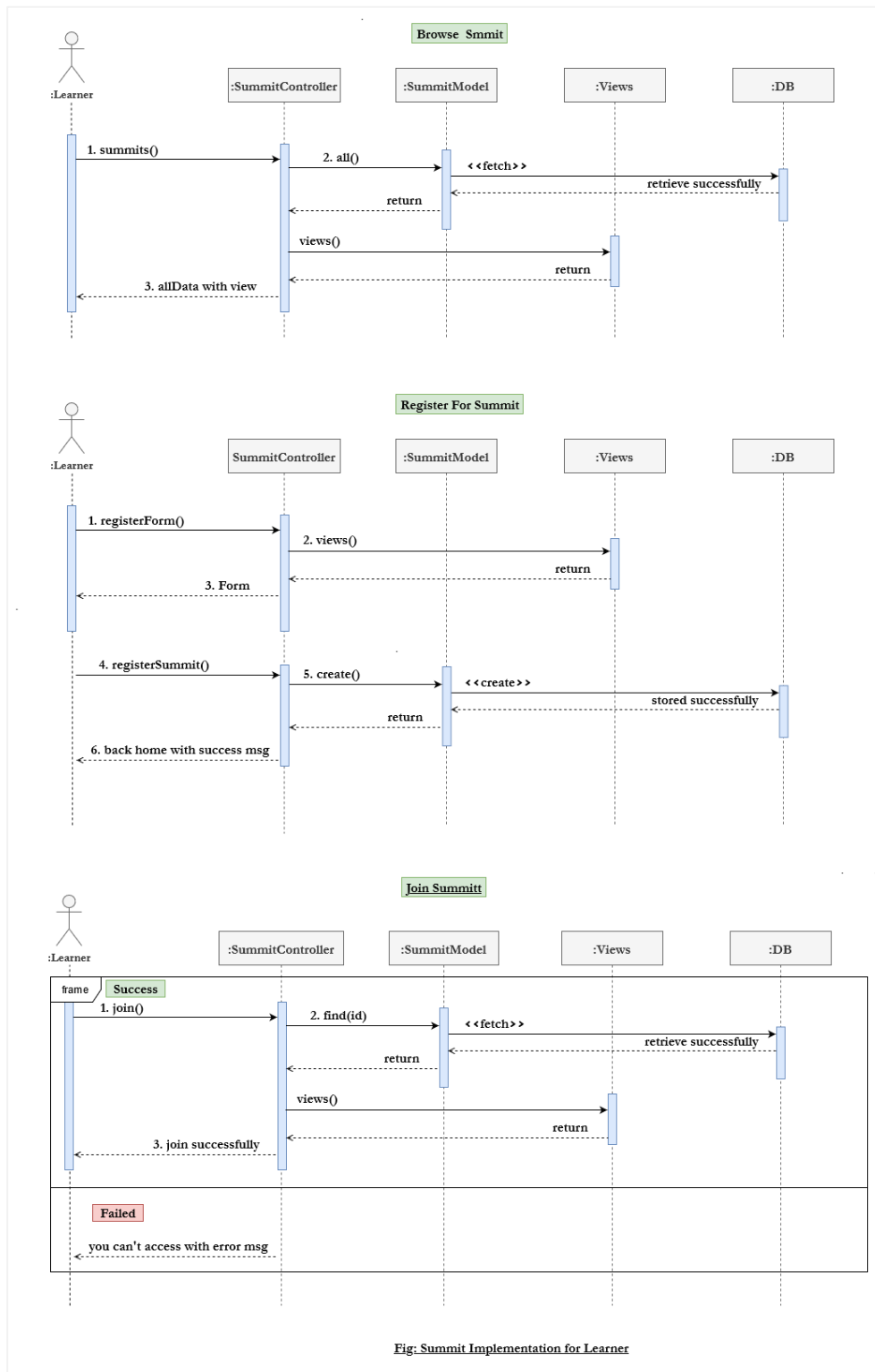
## 6.6 Summit Implementation Sequence Diagram for Trainer & Admin



**NOTE:** For better view please follow the link – [Summit Sequence Diagram](#)

Fig 6.6: Summit Implementation Sequence Diagram

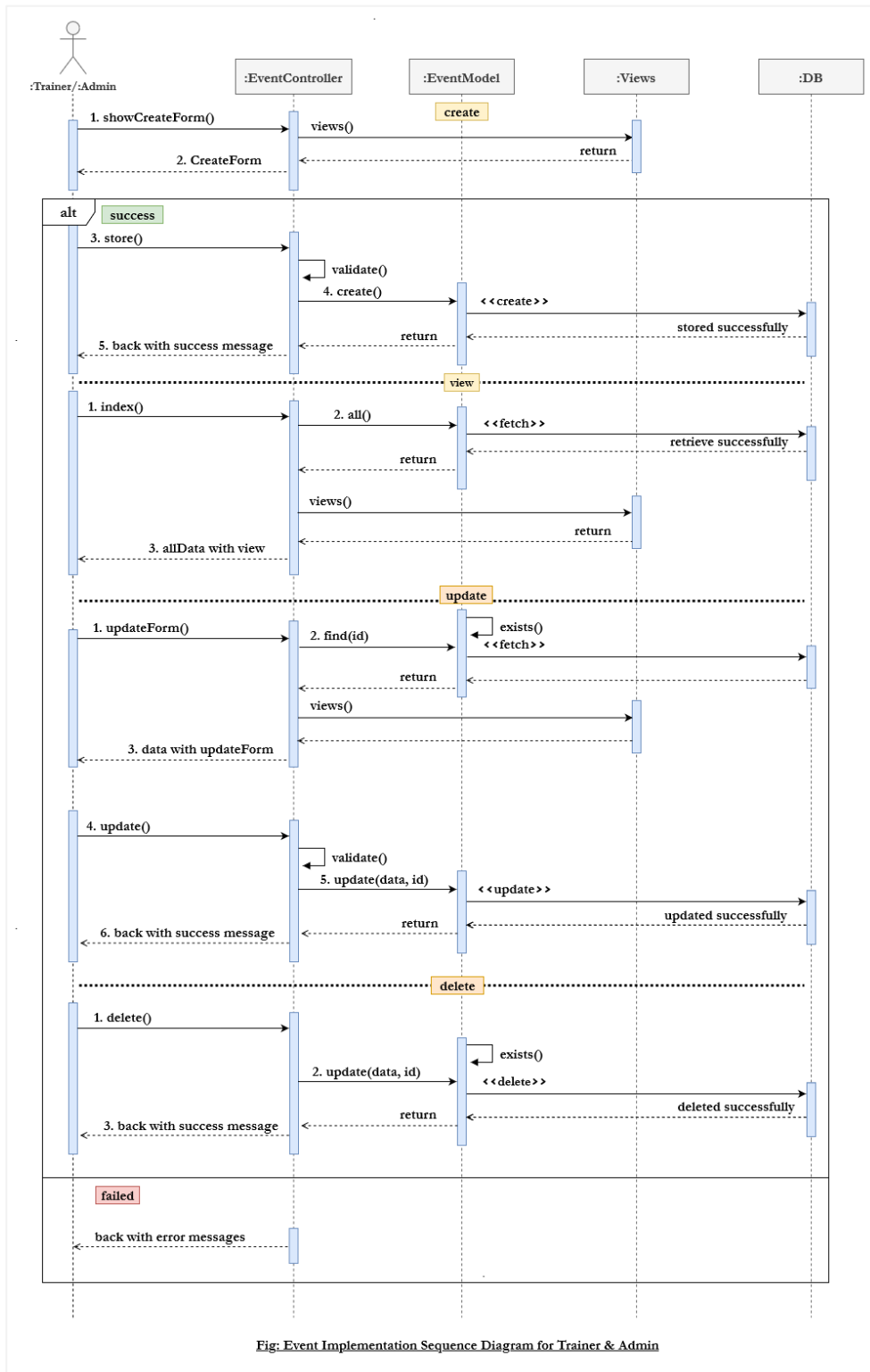
## 6.7 Summit Implementation Sequence Diagram for Learner



**NOTE:** For better view please follow the link – [Summit Sequence Diagram for Learner](#)

Fig: 6.7 Summit Implementation Sequence Diagram

## 6.8 Live Event Implementation Sequence Diagram for Trainer & admin



**NOTE:** For better view please follow the link – [Live Event Diagram for Learner](#)

Fig 6.8: Event Implementation Sequence Diagram

## 6.9 Live Event Implementation Sequence Diagram for Learner

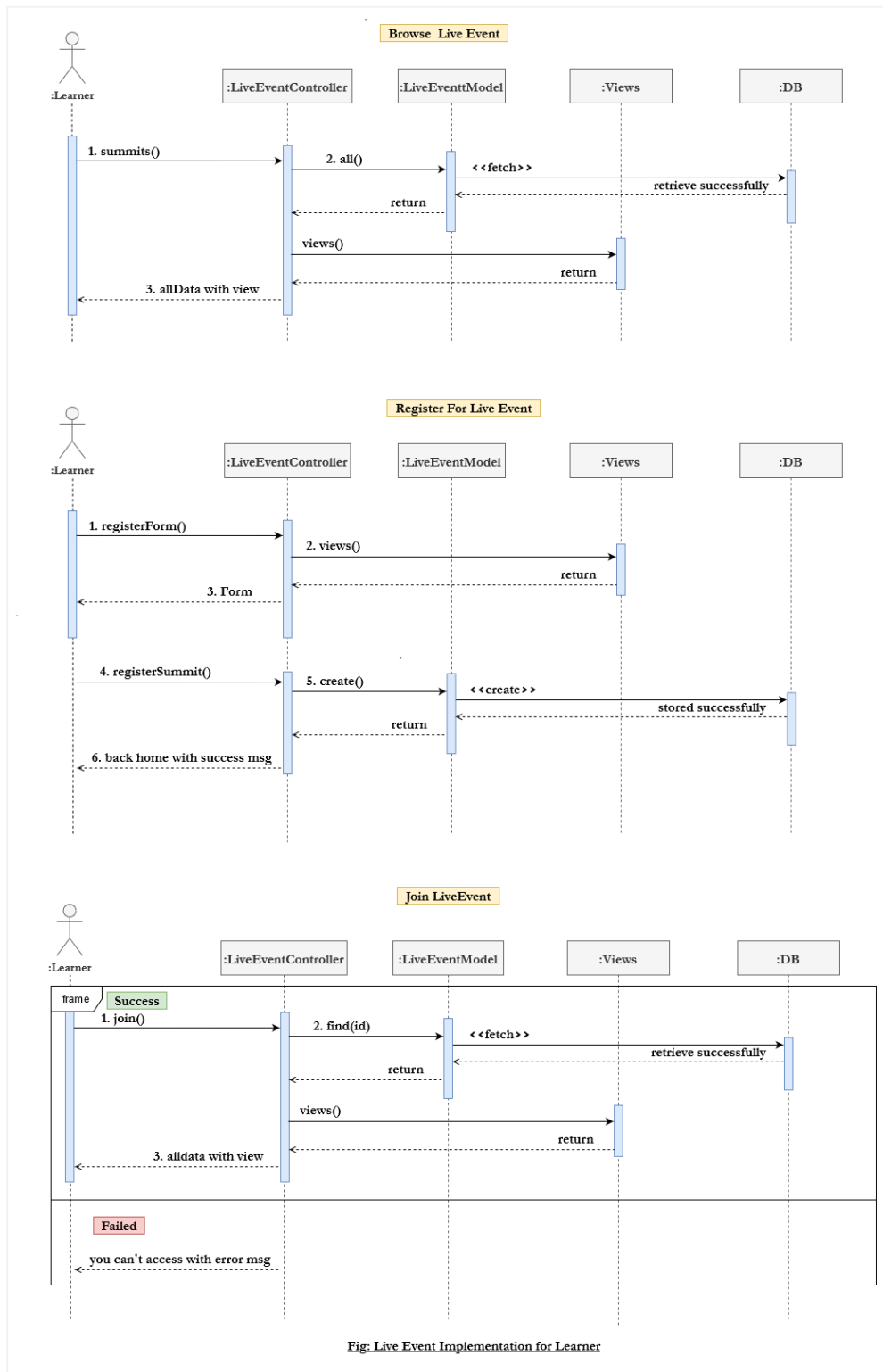
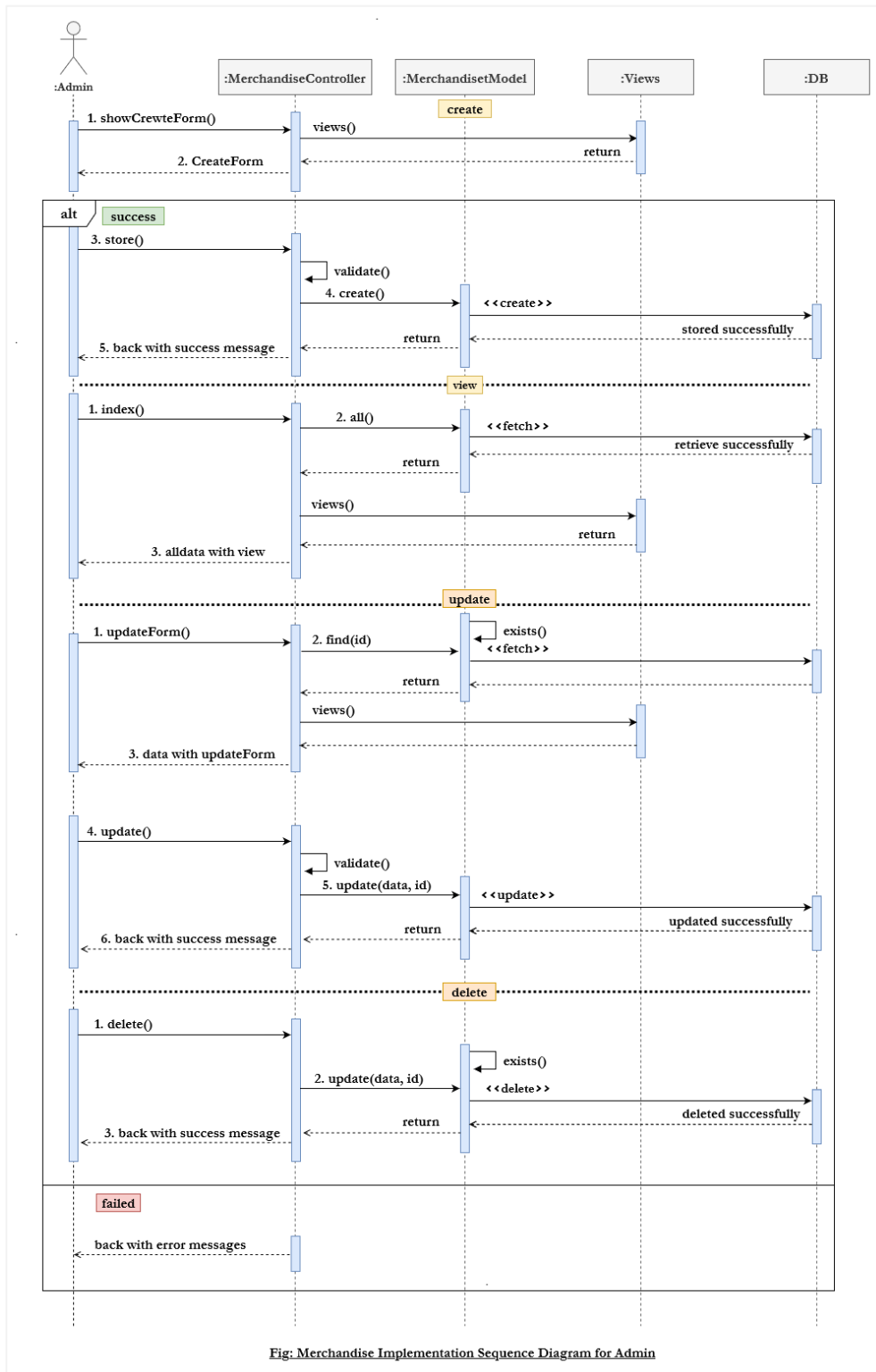


Fig. Live Event Implementation for Learner

**NOTE:** For better view please follow the link – [Live Event Sequence Diagram for Learner](#)

Fig 6.9: Live Event Implementation Sequence Diagram

## 6.10 Merchandise Product Sequence Diagram for Admin

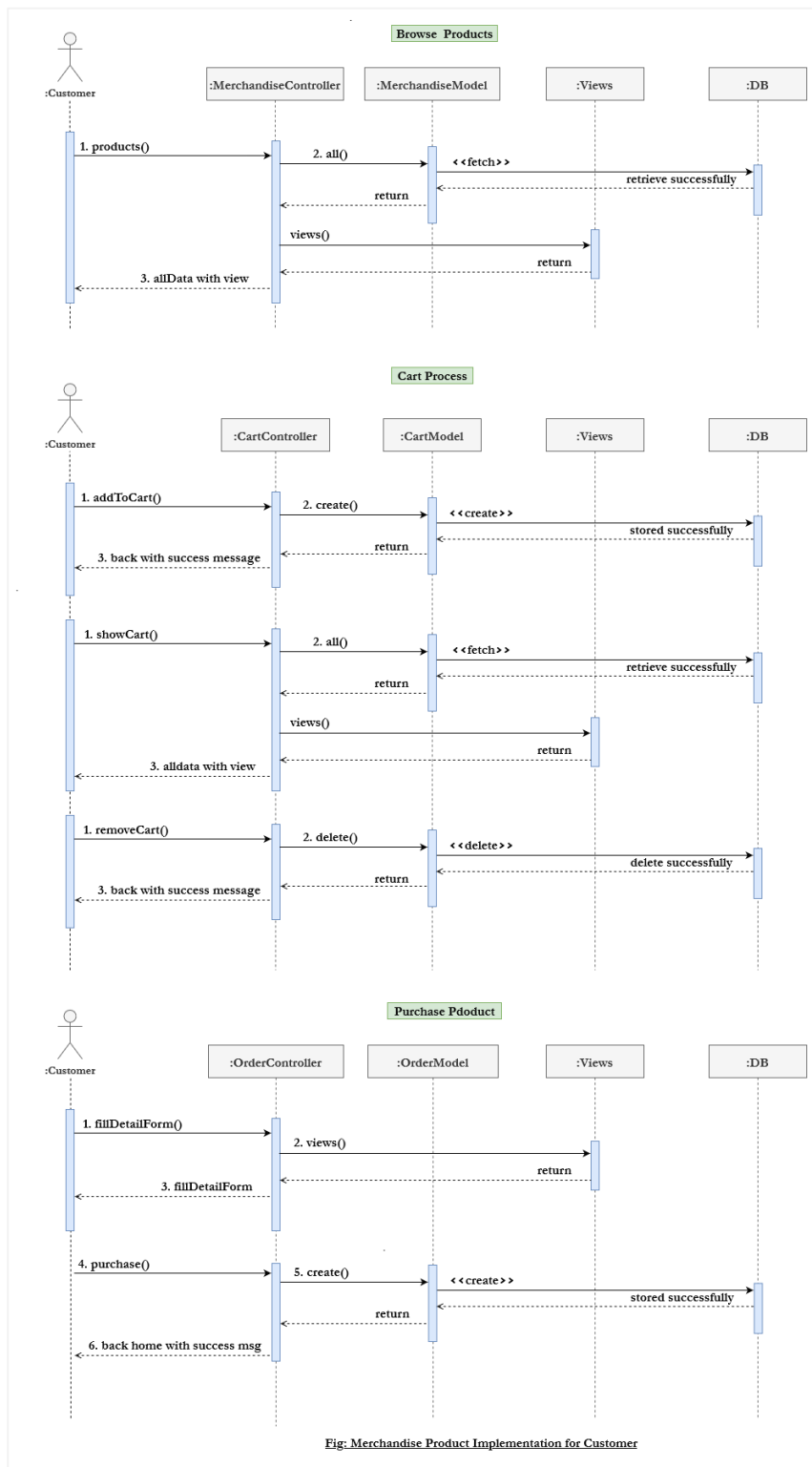


**NOTE:** For better view please follow the link – [Merchandise Product Sequence for Admin](#)

Fig 6.10: Merchandise Product Implementation Sequence Diagram



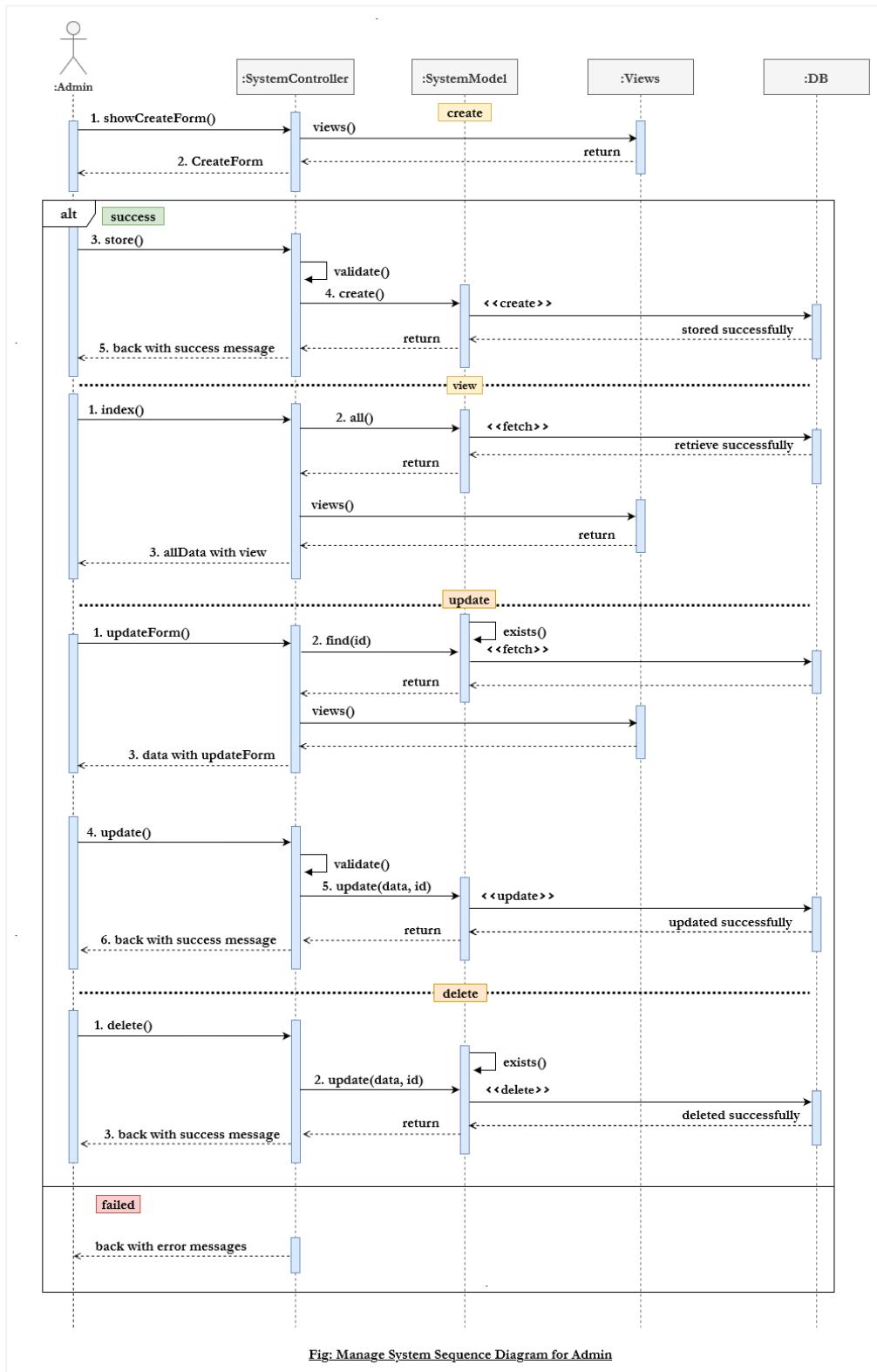
## 6.11 Merchandise Product Sequence Diagram for Customer



**NOTE:** For better view please follow the link – [Merchandise Product Sequence for Customer](#)

Fig 6.11: Merchandise Product Implementation Sequence Diagram

## 6.12 Manage System Sequence Diagram for Admin



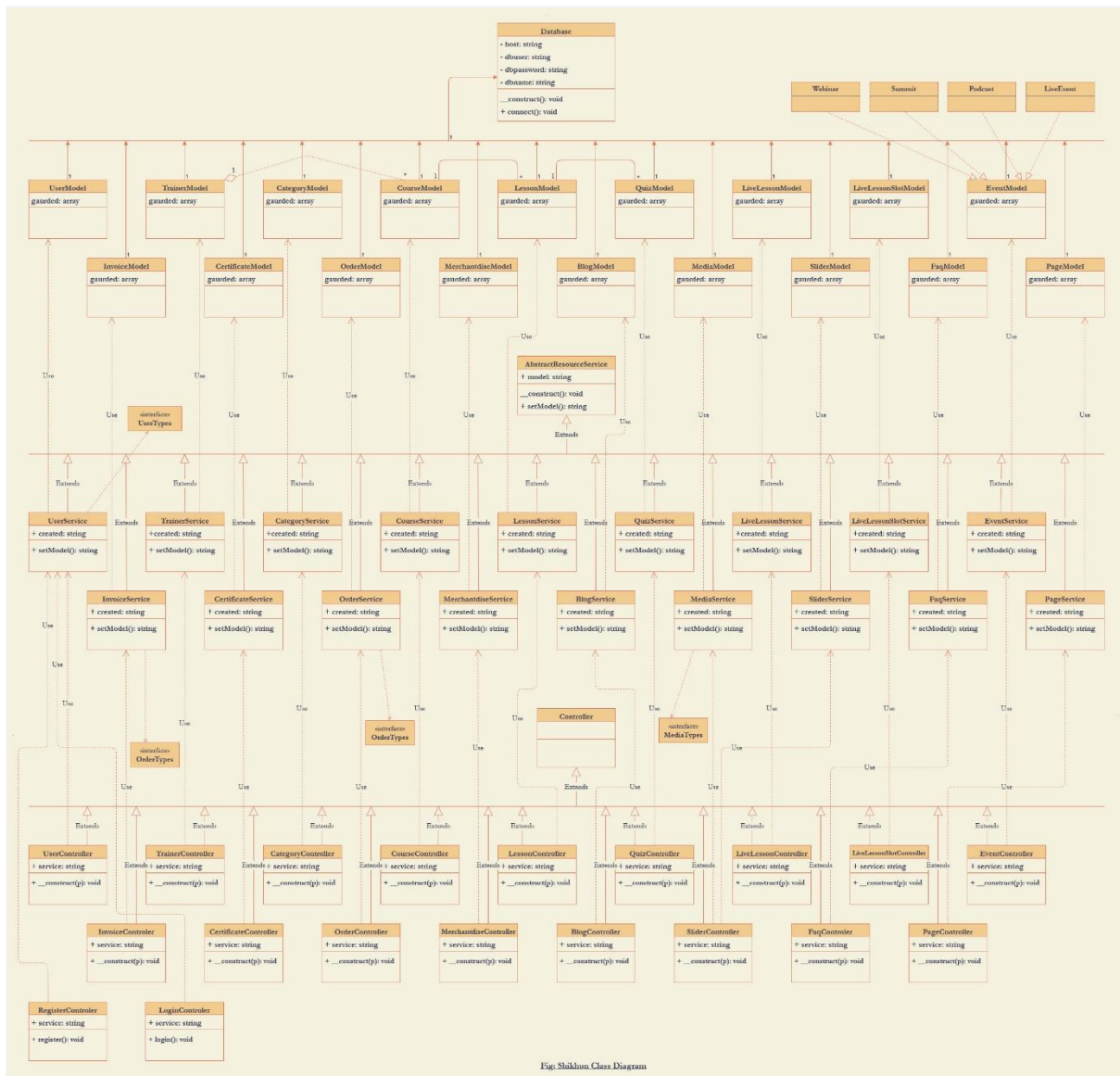
**NOTE:** For better view please follow the link – [Manage System Sequence Diagram](#)

Fig 6.12: Manage System Sequence Diagram



# Chapter 8: Shikhun's Class Diagram

## 8.1 Shikhun's Class Diagram



**NOTE:** For better view please follow the link - [Shikhun's Class Diagram](#)

Fig 8.1: Shikhun's Class Diagram

## 8.2 Detail Design of Classes

### 8.2.1 Database class

Table 8.2.1: Database class

Database Class
- host: string
- dbuser: string
- dbpassword: string
- dbname: string
+ __construct(): void
+ connect(): void

### 8.2.2. AbstractResourceService Class

Table 8.2.2: AbstractResourceService Class

AbstractResourceService Class
+ model: string
+ setModel(): void
+ create(): Object
+ createMultiple(): Object
+ find(primaryKey: int, select: array): Object
+ first(): Object
+ last(): Object
+ latest(): Object
+ all(): Object
+ update(data: array, id: int, column:string): Object
+ delete(id: int): Object
+ updateByModel(model, data: array): Object
+ orderBy(column: string, sorting: string): Object
+ orderBy(column: string, sorting: string): Object
+ orderByDesc(column: string, sorting: string): Object
+ with(relation: array): Object
+ paginate(page: int): Object

### 8.2.3 UserModel, UserService, UserController Class

Table 8.2.3: UserModel, UserService, UserController Class

UserModel Class	UserService Class	UserController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void

### 8.2.4 TrainerModel, TrainerService & TrainerController class

Table 8.2.4: UserModel, UserService, UserController Class

TrainerModel Class	TrainerService Class	TrainerController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ courses(): Object + learners(): Object	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + blockApprove(): void

### 8.2.5 CategoryModel, CategoryService & CategoryController Class

Table 8.2.5: CategoryModel, CategoryService & CategoryController class

CategoryModel Class	CategoryService Classs	CategoryController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ courses(): Model Object	+ setModel(): void +categories(): M. Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + categories(): Object

### 8.2.6 CourseModel, CourseService & CourseController class

Table 8.2.6: CourseModel, CourseService & CourseController Class

CourseModel Class	CourseService Classs	CourseController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ category(): Object + lessons(): Object + teachers(): Object + students(): Object + orderItem(): Object + courseTimeline(): Object + progress(): Object	+ setModel(): void + courses(): Object + teachers(): Object + students(): Object + orders(): Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + courses(): Object + teachers(): Object + students(): Object + orders(): Object

### 8.2.7 LessonModel, LessonService & LessonController Class

Table 8.2.7: LessonModel, LessonService & LessonController Class

LessonModel Class	LessonService Class	LessonController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ course(): Object + tests(): Object + students(): Object + isComplete(): Object	+ setModel(): void + course(): Object + students(): Object + complete(): Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + course(): Object + students(): Object + complete(): Object

### 8.2.8 QuizModel, QuizService & QuizController class

Table 8.2.8: QuizModel, QuizService & QuizController Class

QuizModel Class	QuizService Class	QuizController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ course(): Object + lessons(): Object + questions(): Object + isComplete(): Object	+ setModel(): void + course(): Object + questions(): Object + complete(): Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + course(): Object + questions(): Object + complete(): Object



### 8.2.9 LiveLessonModel, LiveLessonService & LiveLessonController Class

Table 8.2.9: LiveLessonModel, LiveLessonService & LiveLessonController Class

LiveLessonModel Class	LiveLessonService Classs	LiveLessonController Class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ lessons(): Object	+ setModel(): void + lessons(): Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + lessons(): Object

### 8.2.10 LiveLessonSlotModel, LiveLessonSlotService & LiveLessonSlotController class

Table 8.2.10: LiveLessonSlot Model, Service & Controller

LiveLessonSlotModel Class	LiveLessonSlotService	LiveLessonSlotController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ lessonsSlot(): Object + user: object + lessons: object	+ setModel(): void + lessons(): Object	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + lessons(): Object

### 8.2.11 EventModel, EventService & EventSlotController class

Table 8.2.11: EventModel, EventService & EventSlotController Class

EventModel Class	EventService class	EventController class
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): void + show(): void + update(): void + delete(): void + eventRegistartion(): void

### 8.2.12 InvoiceModel, InvoiceService & InvoiceController class

Table 8.2.11: EventModel, EventService & EventSlotController Class

InvoiceModel Class	InvoiceService	InvoiceController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ user(): object + order(): object	+ setModel(): void	+ __construct(service): void + getInvoice(): void + showInvoice(): void

### 8.2.13 CertificateModel, CertificateService & CertificateController Class

Table 8.2.13: Certificate Model, Service & Controller Class

CertificateModel Class	CertificateService	CertificateController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ user(): object + course(): object	+ setModel(): void	+ __construct(service): void + getCertificate(): void + generateCertificate(): void + download: void()

### 8.2.14 OrderModel, OrderService & OrderController Class

Table 8.2.14: Certificate Model, Service & Controller Class

InvoiceModel Class	InvoiceService	InvoiceController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
+ user(): object + item(): object + invoice: object	+ setModel(): void	+ __construct(service): void + index(): void + create(): void + getData(): object + complete: void()

### 8.2.15 MerchandiseModel, MerchandiseService & MerchandiseController class

Table 8.2.15: Certificate Model, Service & Controller Class

MerchandiseModel Class	MerchandiseService	MerchandiseController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): object + show: void() + update(): void + delete(): void

### 8.2.16 BlogModel, BlogService & BlogController class

Table 8.2.16: BlogModel, BlogService & BlogController Class

BlogModel Class	BlogService	BlogController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): object + show: void() + update(): void + delete(): void

### 8.2.17 FaqModel, FaqService & FaqController Class

Table 8.2.17: FaqModel, FaqService & FaqController Class

FaqModel Class	FaqService	FaqController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): object + show: void() + update(): void + delete(): void

### 8.2.18 PageModel, PageService & PageController Class

Table 8.2.17: PageModel, PageService & PageController Class

PageModel Class	PageService	PageController
+ gaurded: array	+ created: const + updated: const + deleted: const	+ service: string
	+ setModel(): void	+ __construct(service): void + index(): void + add(): void + create(): object + show: void() + update(): void + delete(): void

## Chapter 9: System Testing

---

### 9.1 Testing Features

1. Registration
2. Course, Lesson, Merchandise Product, Events Create, Read, Update, delete
3. Course show on user end
4. Course Purchase & Process
5. Lesson Process
6. System Management

### 9.2 Testing Approach

1. The whole system will be test manually step by step
2. User acceptance will be provide for the system test
3. Admin activities will be test for entire system

### 9.3 Pass/Fail Criteria

1. Component Pass/Fail criteria – The test will pass if the case meet the object designed requirements of fail if not
2. Integration Pass/Fail criteria – The test will pass if the case meets the object design architecture of fail if not.

### 9.4 Suspension and Resumption

1. Regression Testing - The system should work properly after each change on the system.
2. Database change - The system will not work properly if we change database name.
3. Build Acceptance Test - The system will test after every build
4. System Design - The system should work properly after each change in the design.

### 9.5 Testing Schedule

Table 9.5: Testing Schedule (Developer Form)

Test Phase	Times
Test Plan Creation	1 weeks
Test specification creation	1 weeks
Test specification team review	1 weeks
Component testing	3 weeks
Integration Testing	1 weeks
System Testing	2 weeks

## 9.6. Test Cases

### 9.6.1 Test Case 01:

Registration

Test Case: 01	Test Case Name: Registration
Component: Registration	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 20-03-2021
Executed By: Amin Hossain	Execution Date: 21-03-2021

Table 9.6.1: User Registration

Step	Action	Pass/Fail	Comment
1	Click on register button	Pass	N/A
2	View register page	Pass	N/A
3	Fill up the register form and missed any required filed	Pass	Please fill up all the required fields
4	Fill up all the required fields and submit form	Pass	Registered successfully

### 9.6.2 Test Case 02:

Course, Lesson, Merchandise Product, Summit, Live Event Create

Test Case: 02	Test Case Name: Create
Component: Create Functionality	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 22-03-2021
Executed By: Amin Hossain	Execution Date: 24-03-2021

Table 9.6.2: Course, Lesson, Products, Summit, Live Event Create

Step	Action	Pass/Fail	Comment
1	Click on create button	Pass	N/A
2	View the requested page	Pass	N/A
3	Fill up the form and missed any required filed	Pass	Please fill up all the required fields
4	Fill up all the required fields and submit form	Pass	Created successfully

### 9.6.3 Test Case 03:

Course, Lesson, Merchandise Product, Summit, Live Event Read

Test Case: 03	Test Case Name: Read
Component: Read Functionality	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 22-03-2021
Executed By: Amin Hossain	Execution Date: 24-03-2021

Table 9.6.3: Course, Lesson, Products, Summit, Live Event Read

Step	Action	Pass/Fail	Comment
1	Click on module button	Pass	N/A
2	View the requested page	Pass	N/A

### 9.6.4 Test Case 04:

Course, Lesson, Merchandise Product, Summit, Live Event Update

Test Case: 04	Test Case Name: Update
Component: Update Functionality	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 25-03-2021
Executed By: Amin Hossain	Execution Date: 25-03-2021

Table 9.6.4: Course, Lesson, Products, Summit, Live Event Update

Step	Action	Pass/Fail	Comment
1	Click on update button	Pass	N/A
2	View the requested page	Pass	N/A
3	Fill up the form and missed any required filed	Pass	Please fill up all the required fields
4	Fill up all the required fields and submit form	Pass	Updated successfully



### 9.6.5 Test Case 05:

Course, Lesson, Merchandise Product, Summit, Live Event Delete

Test Case: 05	Test Case Name: Delete
Component: Delete Functionality	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 28-03-2021
Executed By: Amin Hossain	Execution Date: 30-03-2021

Table 9.6.5: Course, Lesson, Products, Summit, Live Event Delete

Step	Action	Pass/Fail	Comment
1	Click on delete button	Pass	N/A
2	Click the delete button and found internal error (500)	Pass	Something went wrong please try again later

### 9.6.6 Test Case 06:

Courses show on users end

Test Case: 06	Test Case Name: Show courses
Component: Course	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 05-04-2021
Executed By: Amin Hossain	Execution Date: 06-04-2021

Table 9.6.6: Courses show on users end

Step	Action	Pass/Fail	Comment
1	Click on courses menu	Pass	N/A
2	View the courses page	Pass	N/A
3	Click on courses menu & Internal error found (500)	Pass	Sorry! No courses found.

### 9.6.7 Test Case 07:

#### Courses Purchase & Process

Test Case: 07	Test Case Name: Course Purchase & Process
Component: Course	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 10-04-2021
Executed By:	Execution Date:

Table 9.6.7: Courses Purchase & Process

Step	Action	Pass/Fail	Comment
1	Click on add to cart icon		Course added successfully
2	Go to checkout page		N/A
3	Fill all the required details		N/A
4	Missed any required field and submit		Please fill all the required fields
5	Choose payment type		N/A
6	No payment method choose		Please choose a payment method
7	Submit the information		Course purchase successfully

### 9.6.8 Test Case 08:

#### Lesson Process

Test Case: 08	Test Case Name: Lesson Process
Component: Lesson	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 15-04-2021
Executed By:	Execution Date:

Table 9.6.8: Lesson Process

Step	Action	Pass/Fail	Comment
1	Enter into course		N/A
2	Start lesson		N/A
3	Start lessons & no lesson able to load		Lessons not added yet.

### 9.6.9 Test Case 09:

#### System Management

Test Case: 09	Test Case Name: System Management
Component: System Management	Sub Component: N/A
Design By: Amin Hossain(172-35-2180)	Design Date: 20-04-2021
Executed By:	Execution Date:

Table 9.6.9: System Management

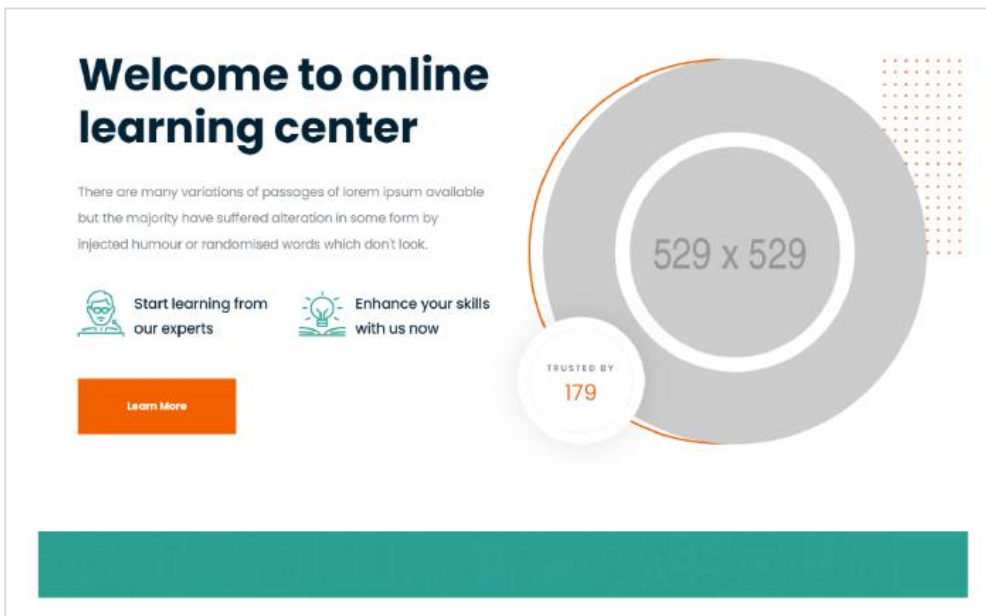
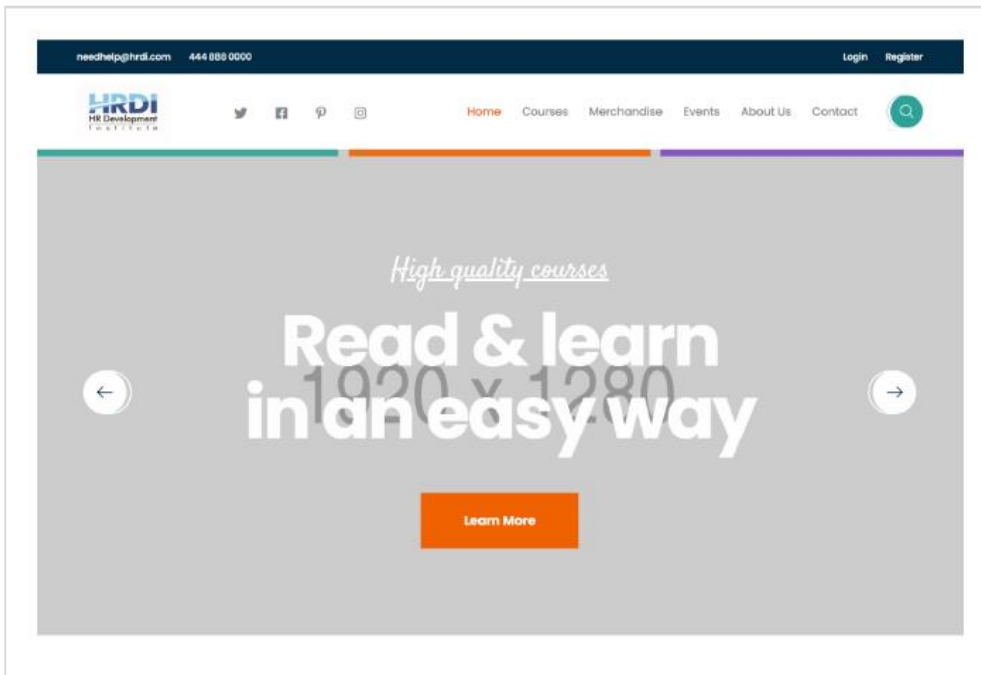
Step	Action	Pass/Fail	Comment
1	Click on required module		N/A
2	Manage the module		N/A
3	Click & not able to load module		System under maintenance. Please try again later.

## Chapter 10: User Manual

### 10.1 User Manual (Frontend)


**NOTE:** For Full User Manual Please follow the link - [User Manual](#)

#### 10.1.1 Home Page



.....

## Browse online course categories




 Photography	 Marketing	 Health & Fitness	 IT & Software	 Development	 Music
--	--	---	--	--	---

[View All Categories](#)

### Our company & partners

.....

## Our latest news & articles

 by Admin . 3 Comments <b>Get a tips to develop a quality education</b>	 by Admin . 3 Comments <b>Learn variety of programs and courses</b>	 by Admin . 3 Comments <b>Summer high school journalism camp</b>
--	--	---

**We've best teachers in every subject**

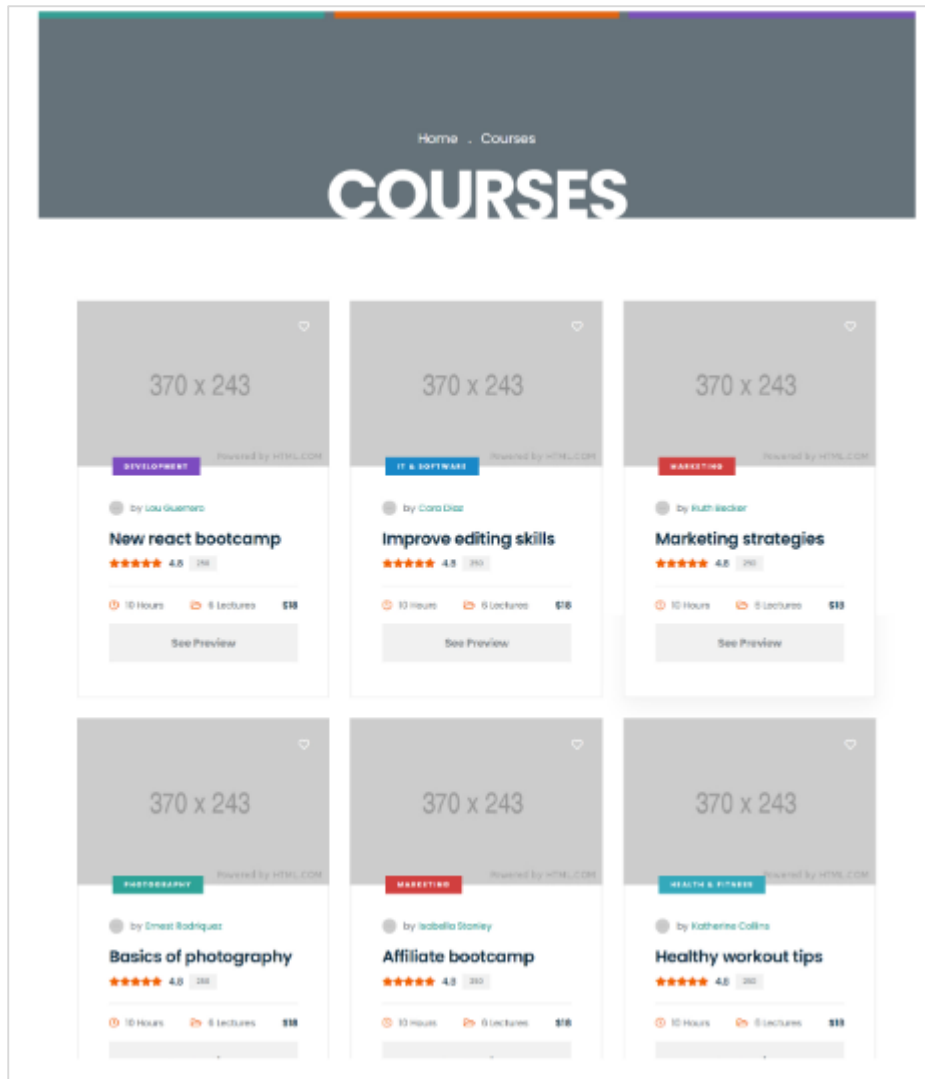
Photography, Music, Math, English, Science

Get latest courses updates by signing up

Enter your email  [Subscribe](#)

<p><b>Course</b></p> <p>Introduction Web Design <b>Mike Hardson</b></p> <p>Learning MBA Management <b>Jessica Brown</b></p>	<p><b>Explore</b></p> <p>About Overview Teachers Join Us Our News</p> <p>Help Contact Register Now</p>	<p><b>Gallery</b></p> <table border="1"> <tr> <td>Image</td> <td>Image</td> <td>Image</td> </tr> <tr> <td>Image</td> <td>Image</td> <td>Image</td> </tr> </table>	Image	Image	Image	Image	Image	Image	<p><b>About</b></p> <p>&gt;Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin euismod sagittis euismod.</p> <p><a href="#">Contact</a> <a href="#">Purchase</a></p>
Image	Image	Image							
Image	Image	Image							

## 10.1.2 Courses page



### 10.1.3 Course Details Page

Home . Course Details

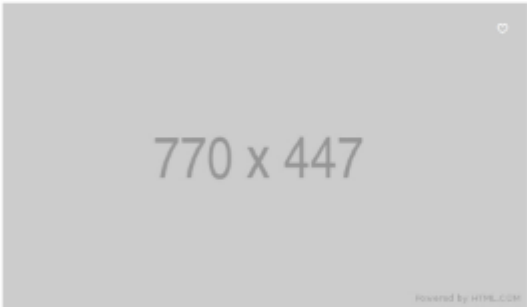
# COURSE DETAILS

by **Addie Walters**

## Improve editing skills

★★★★★ 4.8 (15)

MARKETING



Powered by HTML5.COM

Overview | Curriculum | Reviews

Aelles port lacus quis enim vix sed efficitur turpis gilla sed sit Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus Freehand that replaced versions of Lorem ipsum.

COURSE PRICE

**\$18.00**

Buy This Course

Duration: 10 hours

Lectures: 6

Students: Max 4

Video: 8 hours

Skill Level: **Advanced**

Language: **English**

**New Courses**


by **Lydia Byrd**  
Marketing strategies  
★★★★★ 4.8



## 10.1.4 Contact Us


Home . Contact

# CONTACT




### About Us

Lorem ipsum is simply free text available in the market to use deliver satisfaction.



### Our Address

855 Road Brooklyn Street,  
600 New York, United States of  
America



### Contact info

needhelp@kspau.com  
444 888 0000

## Get in touch with us

Your Name	Email Address
Write Message	
<input type="submit" value="Submit Comment"/>	

## 10.1.5 Admin & Trainer Panel

The screenshot displays the HRDI Admin & Trainer Panel. At the top left is the HRDI logo and a search bar. The top right shows a home icon, the text 'Admin', and a user profile icon. A left sidebar contains a menu with the following items: Dashboard, Teachers, Orders, Course Categories, Course Management, Course Bundles, Reports, Coupons, Events, Merchandise, Account, and Contact Messages. Below these is a 'SYSTEM' section with Site Management, Access, Settings, and Calendar. The main content area features a large grey box with the text 'Welcome Back Admin'. At the bottom of this area is the copyright notice '© 2021 HR Development Institute'.

## Chapter 11: Tools & Technologies

---

- ✓ **Frontend Technologies**
  - HTML5/CSS3
  - Bootstrap (CSS Framework)
  - Vue.js (JavaScript Framework)
  
- ✓ **Backend Technologies**
  - PHP, Laravel (Framework)
  - MySQL Database
  - MVC Architecture (Repository Design Pattern)
  
- ✓ **Tools**
  - Visual Studio Code (Code Editor)
  - GitHub (Version Control)
  - Trello (Project Management Tool)
  - XAMPP (local env. Development server)

## Chapter 12: Project Important Links

---

Links of Project:

- ✓ **GitHub Link**
  - <https://github.com/shakilanwar1998/shikhun-frontend>
  - <https://github.com/shakilanwar1998/shikhun-backend>
  
- ✓ **Trello Link**
  - <https://trello.com/b/REtgSJQT/shikhun>

### 13.1 Limitations

- Video or Audio call streaming is not available
- Real-time chat facilities is not available

### 13.2 Obstacles & Achievements

Technologies are upgrading day by day. We also planned this software architecture to keep that in our mind as a result till now we don't face any major obstacles throughout the project.

If we talk about the achievements, we think we achieve so many achievements through this project. I would like to mention some of them are, we are able to work on industry level project, we feel proud and honored that we able to work closely with our respected supervisor and mentor **Mr. Md. Anwar Hossen**, able to work for our Software Engineering Department etc.

### 13.3 Future Scope

As this is the beta version for initial release we have some plans for future work. Some are mentioned below-

- Statistics of each learner through data analysis
- AI integration for smart suggestion according to learner progress
- Advance UI for easy interaction
- Integrate internal audio and video streaming
- Integrate internal Real-time Chat facilities for all type of users
- Play & Learn (Shikhun's Special)
- Daily Tips & Tricks
- Popular International Payment System Integration
- Shikhun's for International
- Release for Mobile OS

## REFERENCES

---

### [1]. A SIMPLE EXAMPLE OF OBJECT-ORIENTED DESIGN: AN ADDRESS BOOK

**In-text:** (Cs.gordon.edu, 2020)

**You're Bibliography:** Cs.gordon.edu. (2020). *A Simple Example of Object-Oriented Design: An Address Book*. [online] Available at:

<http://www.cs.gordon.edu/courses/cs211/AddressBookExample/> [Accessed 2 Feb. 2020].

### [2]. LARAVEL

Laravel, "Laravel Documentation", <https://laravel.com/docs/7.x/installation>

### [3]. VUE.js

VUE.js, "vue.js Documentation", <https://vuejs.org/v2/guide/>

### [4]. REPOSITORY DESIGN PATTERN

Medium, "Repository Design Pattern in Laravel Application",

<https://medium.com/thinkcru/laravel-ioc-and-the-repository-pattern-a6faac7c889b>

Medium, "Build Great Software with the Repository Pattern in Laravel",

<https://shishirthedev.medium.com/repository-design-pattern-in-laravel-application-f474798f53ec>

Dev Genius, "Laravel API — Repository Pattern (Make Your Code More Structured)- The Simple Guide", <https://blog.devgenius.io/laravel-api-repository-pattern-make-your-code-more-structured-the-simple-guide-5b770da766d7>

### [5]. ACTIVITY DIAGRAM - LOGIN OR REGISTER | ACTIVITY DIAGRAM, STATE DIAGRAM, ACTIVITIES

**In-text:** (Pinterest, 2020)

**You're Bibliography:** Pinterest. (2020). *Activity Diagram - Login or Register | Activity diagram, State diagram, Activities*. [online] Available at:

<https://www.pinterest.com/pin/422071796297187289/> [Accessed 19 Feb. 2020].

### [6]. UML TUTORIAL FOR BEGINNERS - DIAGRAMS, EXAMPLES, LECTURE NOTES

**In-text:** (UML Tutorial for Beginners, 2020)

**You're Bibliography:** UML Tutorial for Beginners. (2020). *UML tutorial for beginners - Diagrams, Examples, Lecture notes*. [online] Available at:

<https://www.startertutorials.com/uml/> [Accessed 19 Feb. 2020].

### [7]. UML DIAGRAMS

UML-Diagrams, "The Unified Model Language", <https://www.uml-diagrams.org/>

# PLAGIARISM REPORT

6/24/2021

Turnitin

## Turnitin Originality Report

Processed on: 2021년 06월 24일 10:25 +06  
ID: 1611409589  
Word Count: 7782  
Submitted: 1

172-35-2180 By Md. Amin Hossain

Similarity Index

14%

### Similarity by Source

Internet Sources: 9%  
Publications: 1%  
Student Papers: 12%

5% match (student papers from 09-Apr-2018)  
Class: April 2018 Project Report  
Assignment: Student Project  
Paper ID: [943600277](#)

4% match (student papers from 28-Jan-2021)  
[Submitted to Deptford Township High School on 2021-01-28](#)

1% match (Internet from 15-Mar-2020)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3553/P13659%20%2829%25%29.pdf?isAllowed=y&sequence=1>

1% match (Internet from 01-Apr-2020)  
<https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report>

< 1% match (student papers from 31-Mar-2018)  
Class: Article 2018  
Assignment: Journal Article  
Paper ID: [939021081](#)

< 1% match (Internet from 06-Jan-2020)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 07-Apr-2021)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5314/172-15-9923%20%2810%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 12-Jan-2021)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4846/153-15-588%3D%2019%25.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 20-Dec-2019)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3545/P13643%20%2816%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 01-Jun-2021)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5376/162-15-8175%20%2816%29.pdf?sequence=1>

< 1% match (student papers from 01-Jun-2010)  
[Submitted to University of Greenwich on 2010-06-01](#)

< 1% match (student papers from 25-Nov-2016)  
[Submitted to University of Greenwich on 2016-11-25](#)

< 1% match (student papers from 05-Dec-2017)  
[Submitted to University of Greenwich on 2017-12-05](#)

< 1% match (student papers from 29-Mar-2021)

[https://www.turnitin.com/newreport\\_printview.asp?eq=1&eb=1&esm=10&oid=1611409589&sid=0&n=0&m=2&svr=57&r=59.02435445631837&lang=e...](https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=10&oid=1611409589&sid=0&n=0&m=2&svr=57&r=59.02435445631837&lang=e...) 1/11

# ACCOUNCE CLEARANCE

