

System and Software Architecture Description (SSAD)

Sharethetraining.com

Team 11

Xiaoting Cai - Project Manager, LCP

Junfeng Wu - Operational Concept Eng, SA

Yi Ding - Feasibility Analyst

Shubham Gaur - Requirements Engineering, LCP

Dongxue Wang - Prototyper

Chao Lin - Operational Concept Eng, SA

Patrick Horng - IIV&V

10/22/2014

Version History

Date	Author	Version	Changes made	Rationale
10/10/14	Team 11	1.0	<ul style="list-style-type: none">Initial version	<ul style="list-style-type: none">Initial version for Core Foundations Commitment Package
10/16/14	Team 11	1.1	<ul style="list-style-type: none">Modify the system context diagram, artifacts & information diagram and process diagram	<ul style="list-style-type: none">The initial version have some mistakes
10/20/14	Team 11	2.0	<ul style="list-style-type: none">Update use case diagramComplete system context diagram	<ul style="list-style-type: none">FC Package
10/22/14	Team 11	2.1	<ul style="list-style-type: none">Add the part of 2.1.4--5	<ul style="list-style-type: none">FC Package

Table of Contents

System and Software Architecture Description (SSAD)	i
Version History	ii
Table of Contents	iii
Table of Tables	iv
Table of Figures.....	vi
1. Introduction.....	1
1.1 Purpose of the SSAD	1
1.2 Status of the SSAD	1
2. System Analysis.....	2
2.1 System Analysis Overview.....	2
2.2 System Analysis Rationale.....	15
3. Technology-Specific System Design	16
3.1 Design Overview	16
3.2 Design Rationale.....	22
4. Technology-Specific System Design	24
4.1 Design Overview	24
4.2 Design Rationale.....	30
5. Architectural Styles, Patterns and Frameworks.....	32

Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>2</i>
<i>Table 2: Artifacts and Information Summary</i>	<i>3</i>
<i>Table 3: Process Description – add course.....</i>	<i>4</i>
<i>Table 4: Typical Course of Action – add course</i>	<i>4</i>
<i>Table 5: Alternate Course of Action – add course</i>	<i>5</i>
<i>Table 6: Exceptional Course of Action – add course</i>	<i>5</i>
<i>Table 7: Process Description – delete course</i>	<i>5</i>
<i>Table 8: Typical Course of Action – delete course.....</i>	<i>5</i>
<i>Table 9: Alternate Course of Action – delete course</i>	<i>5</i>
<i>Table 10: Exceptional Course of Action – delete course.....</i>	<i>6</i>
<i>Table 11: Process Description – modify course</i>	<i>6</i>
<i>Table 12: Typical Course of Action – modify course</i>	<i>6</i>
<i>Table 13: Alternate Course of Action – modify course.....</i>	<i>6</i>
<i>Table 14: Exceptional Course of Action – modify course</i>	<i>7</i>
<i>Table 15: Process Description – verify new course</i>	<i>7</i>
<i>Table 16: Typical Course of Action – verify new course.....</i>	<i>7</i>
<i>Table 17: Alternate Course of Action – verify new course.....</i>	<i>7</i>
<i>Table 18: Exceptional Course of Action – verify new course.....</i>	<i>7</i>
<i>Table 19: Process Description – Request new course.....</i>	<i>8</i>
<i>Table 20: Typical Course of Action – Request new course</i>	<i>8</i>
<i>Table 21: Alternate Course of Action – Request new course</i>	<i>8</i>
<i>Table 22: Exceptional Course of Action – Request new course</i>	<i>8</i>
<i>Table 23: Process Description – Update schedule.....</i>	<i>8</i>
<i>Table 24: Typical Course of Action – Update schedule</i>	<i>9</i>
<i>Table 25: Alternate Course of Action – Update schedule</i>	<i>9</i>
<i>Table 26: Exceptional Course of Action – Update schedule</i>	<i>9</i>
<i>Table 27: Process Description – login</i>	<i>9</i>
<i>Table 28: Typical Course of Action - login</i>	<i>9</i>
<i>Table 29: Alternate Course of Action – login.....</i>	<i>10</i>

<i>Table 30: Exceptional Course of Action – login.....</i>	<i>10</i>
<i>Table 31: Process Description – logout</i>	<i>10</i>
<i>Table 32: Typical Course of Action - logout</i>	<i>10</i>
<i>Table 33: Process Description – Setup account.....</i>	<i>11</i>
<i>Table 34: Typical Course of Action - Setup account.....</i>	<i>11</i>
<i>Table 35: Alternate Course of Action – Setup account.....</i>	<i>11</i>
<i>Table 36: Exceptional Course of Action – Setup account</i>	<i>11</i>
<i>Table 37: Process Description – share course</i>	<i>12</i>
<i>Table 38: Typical Course of Action - share course</i>	<i>12</i>
<i>Table 39: Alternate Course of Action – share course.....</i>	<i>12</i>
<i>Table 40: Process Description – Search course.....</i>	<i>12</i>
<i>Table 41: Typical Course of Action - Search course.....</i>	<i>13</i>
<i>Table 42: Alternate Course of Action – Search course</i>	<i>13</i>
<i>Table 43: Exceptional Course of Action – Search course</i>	<i>13</i>
<i>Table 44: Process Description – Register course.....</i>	<i>13</i>
<i>Table 45: Typical Course of Action - Register course.....</i>	<i>14</i>
<i>Table 46: Alternate Course of Action – Register course</i>	<i>14</i>
<i>Table 47: Exceptional Course of Action – Register course.....</i>	<i>14</i>
<i>Table 48: Process Description – Review course</i>	<i>14</i>
<i>Table 49: Typical Course of Action - Review course</i>	<i>14</i>
<i>Table 50: Alternate Course of Action – Review course.....</i>	<i>15</i>
<i>Table 51: Hardware Component Description</i>	<i>18</i>
<i>Table 52: Software Component Description.....</i>	<i>18</i>
<i>Table 53: Supporting Software Component Description.....</i>	<i>18</i>
<i>Table 54: Design Class Description.....</i>	<i>19</i>
<i>Table 55: Hardware Component Description</i>	<i>26</i>
<i>Table 56: Software Component Description.....</i>	<i>26</i>
<i>Table 57: Supporting Software Component Description.....</i>	<i>26</i>
<i>Table 58: Design Class Description.....</i>	<i>27</i>
<i>Table 59: Architectural Styles, Patterns, and Frameworks.....</i>	<i>32</i>

Table of Figures

<i>Figure 1: System Context Diagram</i>	<i>2</i>
<i>Figure 2: Artifacts and Information Diagram</i>	<i>3</i>
<i>Figure 3: Process Diagram</i>	<i>4</i>
<i>Figure 4: Hardware Component Class Diagram</i>	<i>16</i>
<i>Figure 5: Software Component Class Diagram</i>	<i>17</i>
<i>Figure 6: Deployment Diagram.....</i>	<i>17</i>
<i>Figure 7: Design Class Diagram.....</i>	<i>19</i>
<i>Figure 8: Process Realization Diagram—search course</i>	<i>21</i>
<i>Figure 9: Process Realization Diagram—register course</i>	<i>22</i>
<i>Figure 10: Hardware Component Class Diagram</i>	<i>24</i>
<i>Figure 11: Software Component Class Diagram</i>	<i>25</i>
<i>Figure 12: Deployment Diagram.....</i>	<i>25</i>
<i>Figure 13: Design Class Diagram.....</i>	<i>27</i>
<i>Figure 14: Process Realization Diagram—search course</i>	<i>29</i>
<i>Figure 15: Process Realization Diagram—register course</i>	<i>30</i>

1. Introduction

1.1 Purpose of the SSAD

The purpose of this SSAD is to document the results of the object-oriented analysis and design (OOA&D) of the system being developed. The SSAD is used by the builder (programmer) as reference to the system architecture. The system being developed should be faithful to the architecture specified in the SSAD. Furthermore, the SSAD is used by the maintainer and clients to help understand the structure of the system once the proposed system is delivered.

1.2 Status of the SSAD

This SSAD is the forth version in which we add the part of 2.1.3 to 5.

2. System Analysis

2.1 System Analysis Overview

The main purposes of the project, share the training, is to provide a platform, a business course sharing website, to trainers and attendees. Users of the website will include the attendees who take courses, trainers who provide courses and administer who manage the website.

2.1.1 System Context

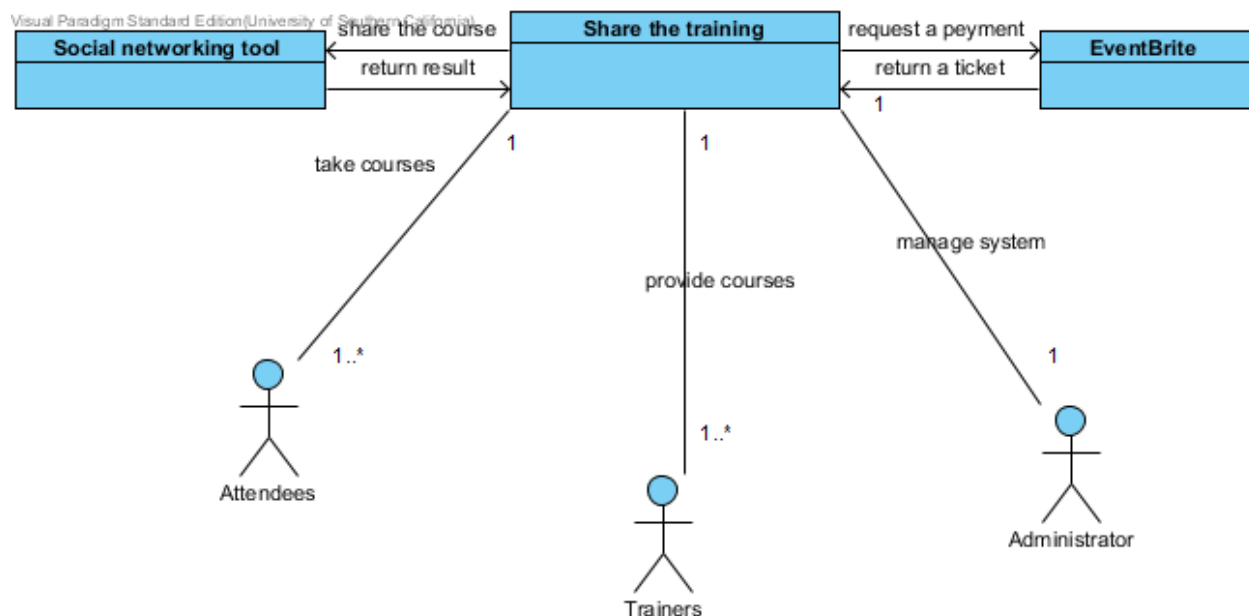


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Administer	Person who review registration and manage the system	Review trainer's request, manage system. Make schedule of the course
Trainers	Person who provide courses	provide courses
Attendees	Person who take courses	register course, share course and review the course

2.1.2 Artifacts & Information

Visual Paradigm Standard Edition (University of Southern California)

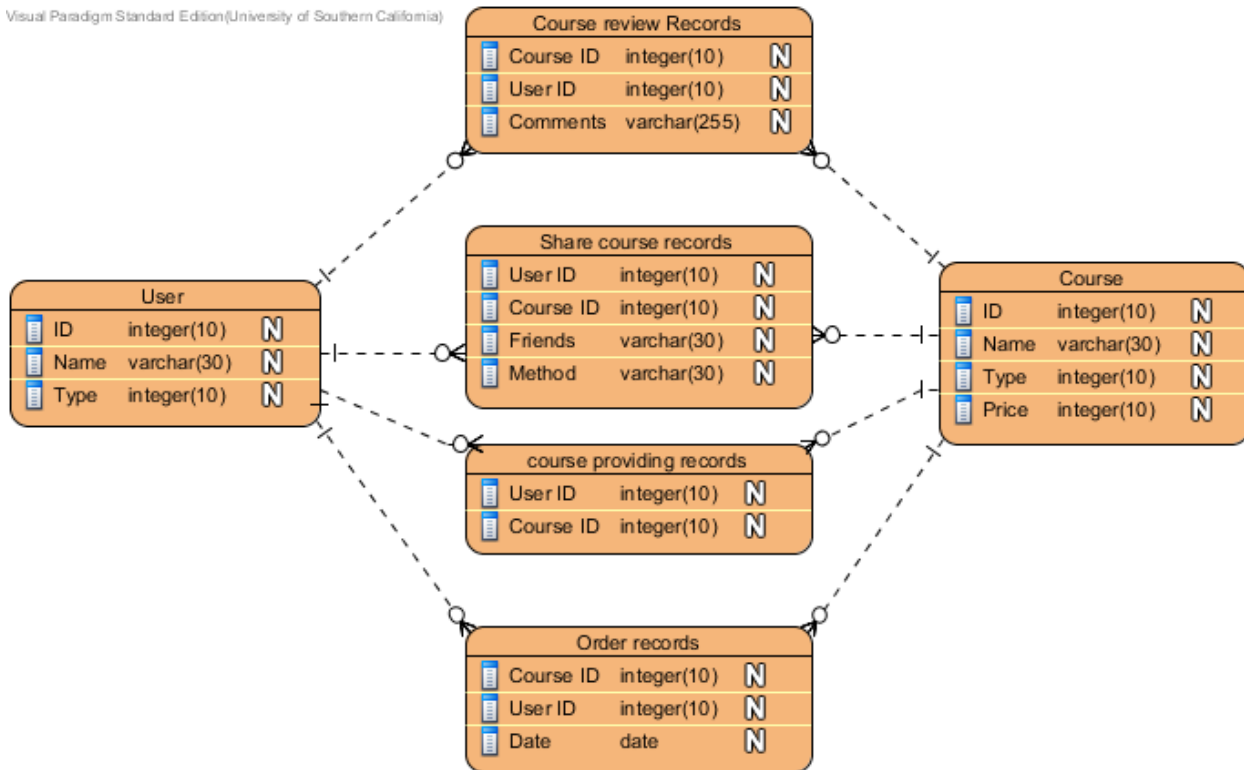


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
ATF-1 User	It contains all the basic info of a user.
ATF-2 Order records	It contains all the orders info, which can help organization to analyze data about customers' choices.
ATF-3 Course providing records	It records the trainers' course providing information
ATF-4 Share course records	It contains all the friends who has been shared coursed by a user. It can help the organization to find the target market
ATF-5 Course review records	It contain rating about courses and trainers, and also course-attendee's feedback. It can largely improve the quality of course, and let the training become transparent.
ATF-6 Course	It contains all information of a course

2.1.3 Behavior

Visual Paradigm Standard Edition (University of Southern California)

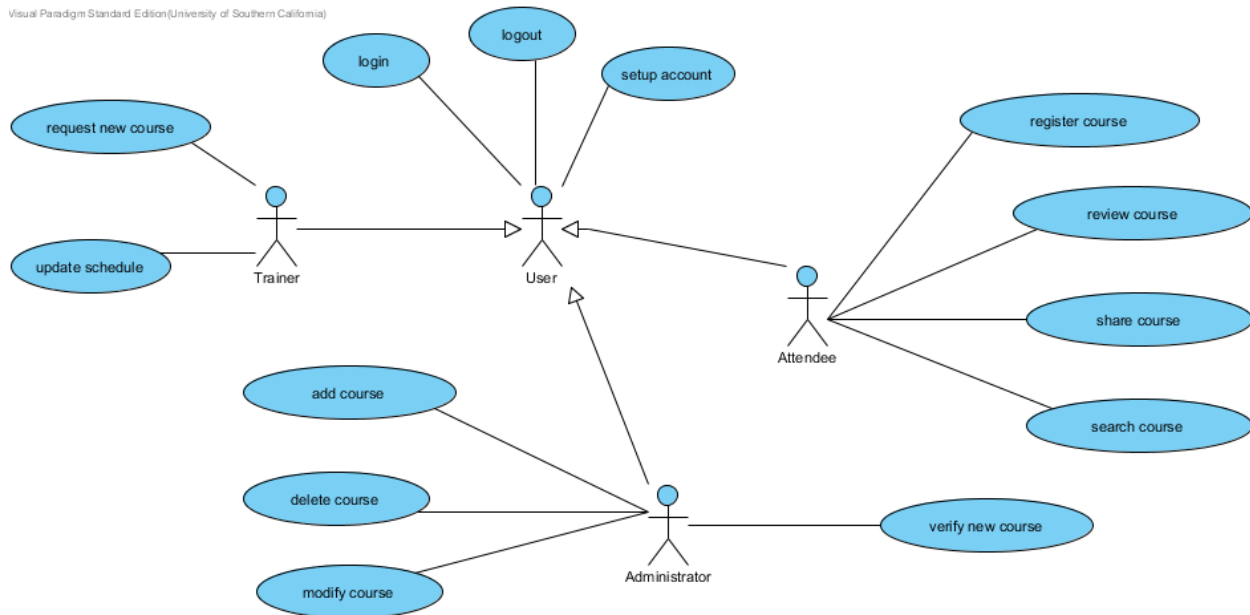


Figure 3: Process Diagram

2.1.3.1 Manage course

2.1.3.1.1 Add course

Table 3: Process Description – add course

Identifier	UC-1: Add course
Purpose	Administrator can add course after verify the course information
Requirements	WC_3117
Development Risks	none
Pre-conditions	Administrator is on the manage course page
Post-conditions	Course database is updated

Table 4: Typical Course of Action – add course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator enters in the course information	
2	Administrator click the “submit”	System save course information into database

Table 5: Alternate Course of Action – add course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator click “back” button	Form is not submitted, return to the former page

Table 6: Exceptional Course of Action – add course

Seq#	Actor's Action	System's Response
1	On the manage course page, administrator enters in some invalid information	
2	administrator click “submit”	System shows invalid information on the fields which is wrong

2.1.3.1.2 Delete course

Table 7: Process Description – delete course

Identifier	UC-2: delete course
Purpose	Administrator can delete course
Requirements	WC_3132
Development Risks	none
Pre-conditions	Administrator is on the manage course page
Post-conditions	Course database is updated

Table 8: Typical Course of Action – delete course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator choose one or some courses	
2	administrator click the “delete”	System delete course information from database

Table 9: Alternate Course of Action – delete course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator click “back” button	Form is not submitted, return to the former page

Table 10: Exceptional Course of Action – delete course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator choose one or some courses which have some attendees	
2	Click the “delete”	System shows confirm box
4	Click “yes”	System delete course information from database and send email to attendees who register this course

2.1.3.1.3 Modify course

Table 11: Process Description – modify course

Identifier	UC-3: modify course
Purpose	Administrator can modify course
Requirements	WC_3117
Development Risks	none
Pre-conditions	Administrator is on the manage course page
Post-conditions	Course database is updated

Table 12: Typical Course of Action – modify course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator modify some course information	
2	administrator click the “submit”	System save course information into database

Table 13: Alternate Course of Action – modify course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator click “back” button	Form is not submitted, return to the former page

Table 14: Exceptional Course of Action – modify course

Seq#	Actor's Action	System's Response
1	On the manage course page, administrator type in some invalid information	
2	Click “submit”	System shows invalid information on the fields which is wrong

2.1.3.2 Verify new course

Table 15: Process Description – verify new course

Identifier	UC-4: verify new course
Purpose	Administrator can verify course
Requirements	WC_3120
Development Risks	none
Pre-conditions	Administrator is on the manage course page
Post-conditions	Course database is updated

Table 16: Typical Course of Action – verify new course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator review the course information which submitted by trainers	
2	Click the “accept”	System save course information into database and send an email to trainer

Table 17: Alternate Course of Action – verify new course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator click “back” button	Form is not submitted, return to the former page

Table 18: Exceptional Course of Action – verify new course

Seq#	Actor's Action	System's Response
1	On the manage course page, administrator think the course is not qualified	
2	Click “reject”	System send an email to trainer

2.1.3.3 Request new course

Table 19: Process Description – Request new course

Identifier	UC-5: Request new course
Purpose	trainer can register course
Requirements	WC_3105
Development Risks	none
Pre-conditions	trainer is on the manage course page
Post-conditions	Course information is send to administrator

Table 20: Typical Course of Action – Request new course

Seq#	Actor's Action	System's Response
1	On the manage course page, the trainer type in all course information	
2	Trainer click the “submit”	System send the course information to administrator

Table 21: Alternate Course of Action – Request new course

Seq#	Actor's Action	System's Response
1	On the manage course page, the administrator click “back” button	Form is not submitted, return to the former page

Table 22: Exceptional Course of Action – Request new course

Seq#	Actor's Action	System's Response
1	On the manage course page, trainer type in some invalid information	
2	Click “submit”	System shows invalid information on the fields which is wrong

2.1.3.4 Update schedule

Table 23: Process Description – Update schedule

Identifier	UC-6: Update schedule
Purpose	trainer can update schedule for administrator to arrange course
Requirements	WC_3107

Development Risks	None
Pre-conditions	trainer is on the request course page
Post-conditions	Course information is send to administrator

Table 24: Typical Course of Action – Update schedule

Seq#	Actor's Action	System's Response
1	On the schedule page, the trainer enters his/her schedule	
2	Trainer click the “submit”	System validate the new schedule and send notification to administrator

Table 25: Alternate Course of Action – Update schedule

Seq#	Actor's Action	System's Response
1	On the schedule page, the trainer click “back” button	Form is not submitted, return to the former page

Table 26: Exceptional Course of Action – Update schedule

Seq#	Actor's Action	System's Response
1	On the schedule page, trainer enters in invalid schedule	
2	trainer click “submit”	System validate the new schedule, and return “invalid schedule”

2.1.3.5 Login

Table 27: Process Description – login

Identifier	UC-7: Login
Purpose	Authenticate a user
Requirements	WC_3123
Development Risks	We need to make sure that the correct privileges for each user are set
Pre-conditions	attendee is not logged on to the system and on Login page
Post-conditions	attendee is signed on to the system with his/her privileges (attendee, trainer or administrator) or login is rejected

Table 28: Typical Course of Action - login

Seq#	Actor's Action	System's Response
------	----------------	-------------------

1	On Login page, User enters username/password correctly	
2	User click the “login”	User is logged into the system and role is changed accordingly

Table 29: Alternate Course of Action – login

Seq#	Actor’s Action	System’s Response
1	On Login page, User clicks on "Back"	User is not logged in and system displays page that was displayed before the login page was displayed

Table 30: Exceptional Course of Action – login

Seq#	Actor’s Action	System’s Response
1	On Login page, User enters username/password	
2	User click “login”	System validates username/password combination and concludes it is incorrect
3		System displays red text on Login page saying that login was incorrect

2.1.3.6 Logout

Table 31: Process Description – logout

Identifier	UC-8: Logout
Purpose	Remove privileges and continue session as unprivileged user
Requirements	WC_3123
Development Risks	none
Pre-conditions	User is logged onto the system and system is displaying a page that needs privileges to be accessed
Post-conditions	User is logged off the system and continues session without privileges

Table 32: Typical Course of Action - logout

Seq#	Actor’s Action	System’s Response
1	On a page that needs privileges	System logs out the privileged account

	to be accessed, user clicks "Log off" link	
--	--	--

2.1.3.7 Setup account

Table 33: Process Description – Setup account

Identifier	UC-9: Setup account
Purpose	User can setup an account, so they can do more things
Requirements	WC_3123
Development Risks	None
Pre-conditions	User is on the sign up page
Post-conditions	User information is saved into database

Table 34: Typical Course of Action - Setup account

Seq#	Actor's Action	System's Response
1	On sign up page, User enters in user information	
2	User click the "sign up"	User's information will be send to server and saved into database

Table 35: Alternate Course of Action – Setup account

Seq#	Actor's Action	System's Response
1	On Sign up page, User clicks on "Back"	Account will not be created

Table 36: Exceptional Course of Action – Setup account

Seq#	Actor's Action	System's Response
1	On sign up page, user enters in invalid information	
2	User click "sign up"	System validates user's information
3		System displays "invalid information"

2.1.3.8 Share course

Table 37: Process Description – share course

Identifier	UC-10: share course
Purpose	Share course information to friends
Requirements	WC_3126
Development Risks	Facebook, Twitter, Linkedin API may be incompatible with the system
Pre-conditions	User on the course list page
Post-conditions	Course information send to friends by email, Facebook, twitter

Table 38: Typical Course of Action - share course

Seq#	Actor's Action	System's Response
1	On the course list, attendee choose one or some courses	
2	attendee click the “share”	
3	Attendee choose a method to share the course information	
4	Attendee choose who will be receive this information	
5	Attendee click “send”	System send the course information to the chosen people

Table 39: Alternate Course of Action – share course

Seq#	Actor's Action	System's Response
1	On the course list page, attendee clicks on "Back"	System displays page that was displayed before
2		System displays red text on Login page saying that login was incorrect

2.1.3.9 Search course

Table 40: Process Description – Search course

Identifier	UC-11: Search course
Purpose	Attendee could search courses which they want to attend
Requirements	WC_3099
Development Risks	none

Pre-conditions	There are some courses in the system
Post-conditions	Course displayed in the course list page

Table 41: Typical Course of Action - Search course

Seq#	Actor's Action	System's Response
1	On the search page, attendee enters in keywords	
2	attendee click the "search"	System use the keywords to search the database and display the course information on the course list page and turn the page to the course list page

Table 42: Alternate Course of Action – Search course

Seq#	Actor's Action	System's Response
1	On search page, attendee clicks on "Back"	system displays page that was displayed before

Table 43: Exceptional Course of Action – Search course

Seq#	Actor's Action	System's Response
1	On the search page, attendee enters invalid keywords	
2	Click "search"	System search the database and found no matches
3		System displays "there is no this course in system"

2.1.3.10 Register course

Table 44: Process Description – Register course

Identifier	UC-12: Register course
Purpose	Attendee could register course which they want to attend
Requirements	WC_3125
Development Risks	Payment gateway may be incompatible with the system
Pre-conditions	Profession is on the course detail page
Post-conditions	Turn page to Eventbrite and finish payment

Table 45: Typical Course of Action - Register course

Seq#	Actor's Action	System's Response
1	On the course detail page, profession click "register"	System turn page to Eventbrite
2	Finish payment	System send a confirmation email to attendee

Table 46: Alternate Course of Action – Register course

Seq#	Actor's Action	System's Response
1	On the course detail page, attendee clicks "Back"	system displays page that was displayed before

Table 47: Exceptional Course of Action – Register course

Seq#	Actor's Action	System's Response
1	On the course detail page, profession click "register"	System turn page to Eventbrite
2	Attendee cannot finish payment	System will hold the course for 1hour

2.1.3.11 Review course

Table 48: Process Description – Review course

Identifier	UC-13: Review course
Purpose	Attendee can rating the course after they attend the course
Requirements	WC_3476
Development Risks	none
Pre-conditions	attendee already attend the course
Post-conditions	Score and comment will be posted on the course detail page

Table 49: Typical Course of Action - Review course

Seq#	Actor's Action	System's Response
1	On the course detail page, attendee give the course a score and write some comments	
2	attendee click the "submit"	System update the course information into the database

Table 50: Alternate Course of Action – Review course

Seq#	Actor's Action	System's Response
1	On course detail page, attendee clicks on "Back"	system displays page that was displayed before

2.1.4 Modes of Operation

The ShareTheTraining website, as we envision implementing it, will operate in only one mode, so nothing further need be said of modes of operation.

2.2 System Analysis Rationale

Based on the analysis of how users will interact with our website, we could define the following 3 kinds of operational stakeholders.

1. Course attendees: This kind of users could search courses, register courses and review and rate courses, which is the main consumer group of ShareTheTraing. The website provide a nice platform for course attendees to search and attend courses near them.
2. Trainers: Every trainers identity should be processed by the administrator of website. Once approved, trainers could submit courses request and enroll new students for their courses. Through the website they have a more convenient way to publicize themselves and improve the number of enrollment.
3. Administrators: Administrator is the admin of the website. They could mange course attendees, trainers, courses, orders and reviews.

There are 2 external system actors that interface with ShareTheTraining. They are:

1. EventBrite - provides payment function to ShareTheTraining.
2. Facebook - provides authentication services and share function for to ShareTheTraining.

3. Technology-Specific System Design

3.1 Design Overview

3.1.1 System Structure

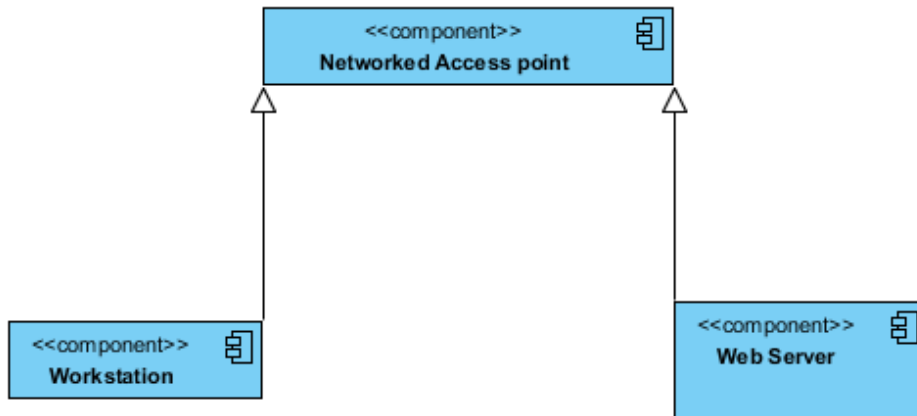


Figure 4: Hardware Component Class Diagram

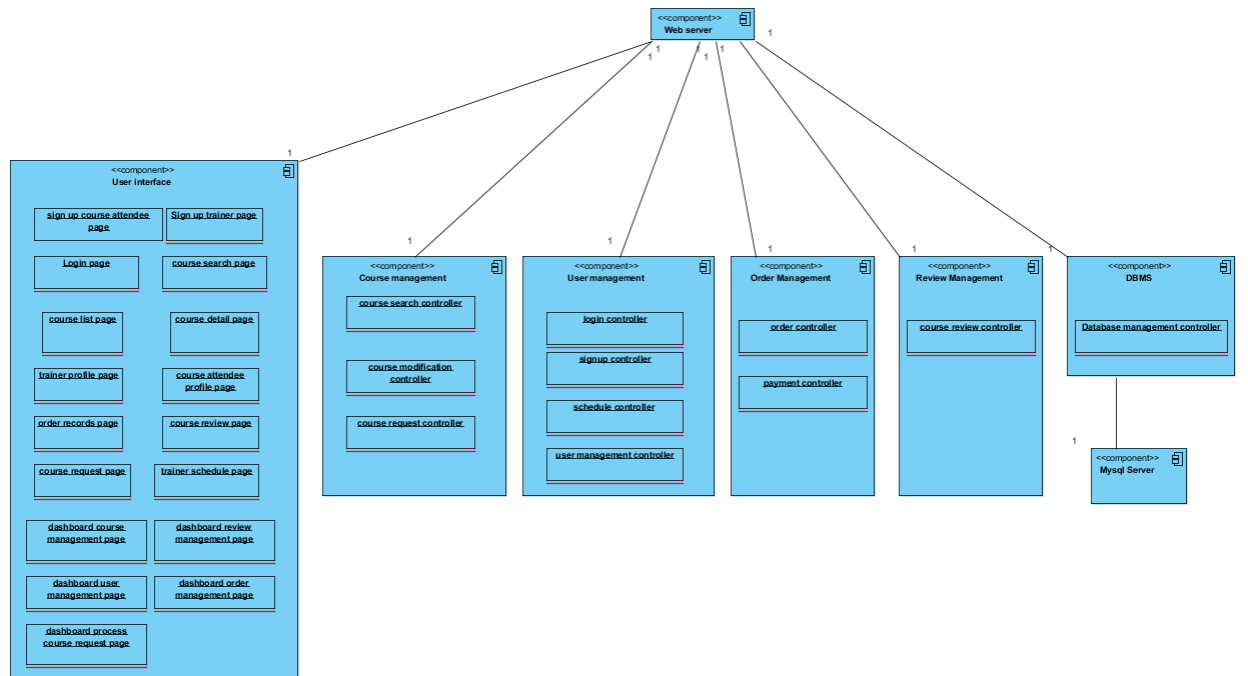


Figure 5: Software Component Class Diagram

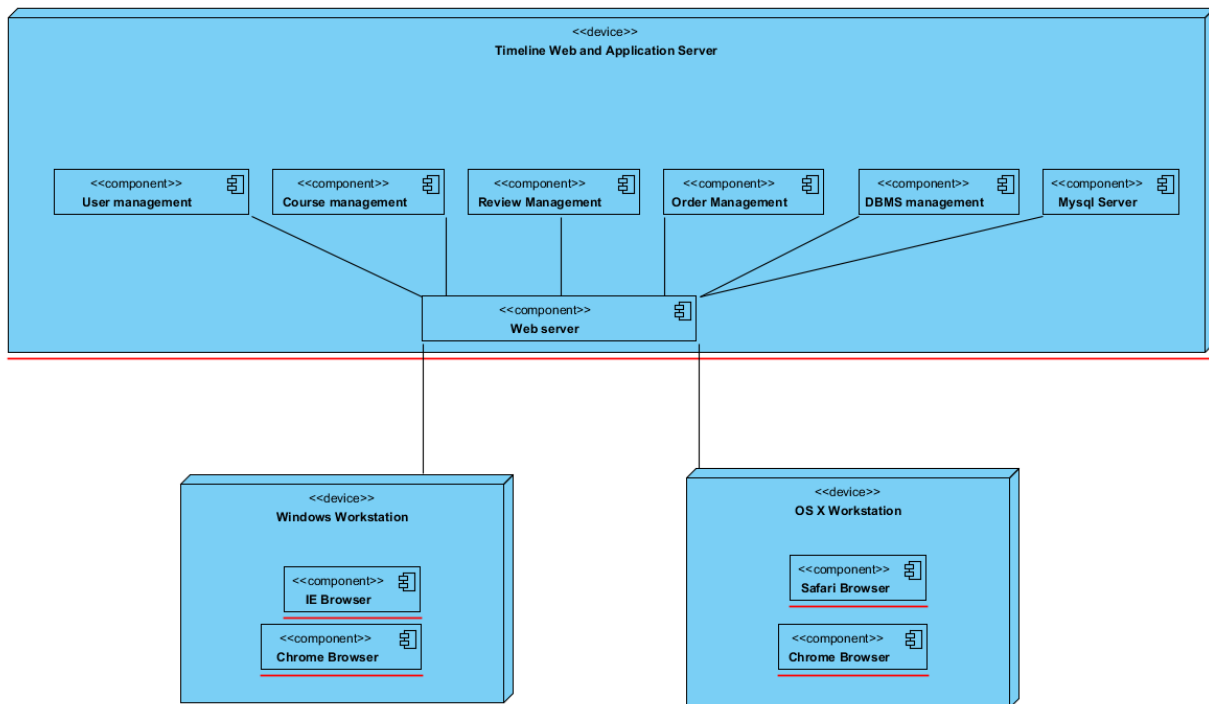


Figure 6: Deployment Diagram

Table 51: Hardware Component Description

Hardware Component	Description
Networked Computer	A computer that is connected to other networked computers through the internet. In our system, every computer will be networked in this manner.
Workstation	A networked computer which is used to access services on the internet. In our system, this will be computers used by people at home or in an office to interact with the ShareTheTraining
Web Server	A networked computer which provides applications to workstations. In our system, this will be the server our software will be deployed on.

Table 52: Software Component Description

Software Component	Description
User Interface Component	This component contains ShareTheTraining pages that users of the system interact with directly
Course Management Component	This component performs all the main functionalities of the course management process. These functionalities include: Approve/ Disapprove courses; Create/ Delete/ Modify courses.
User Management Component	This component performs functions involved in managing information relating to authenticated users of the ShareTheTraining System.
Order Management Component	This component performs functions involved in managing registration and payment of each order.
Review Management Component	This component performs functions involved in managing review and ratings.
DBMS	This is the Database Management System (DBMS) that stores all data used by the CSC Volunteer Tracking System.

Table 53: Supporting Software Component Description

Support Software Component	Description
My SQL	A relational database management system
Play Framework	An open source framework on which build the backend of website

3.1.2 Design Classes

3.1.2.1 User interface

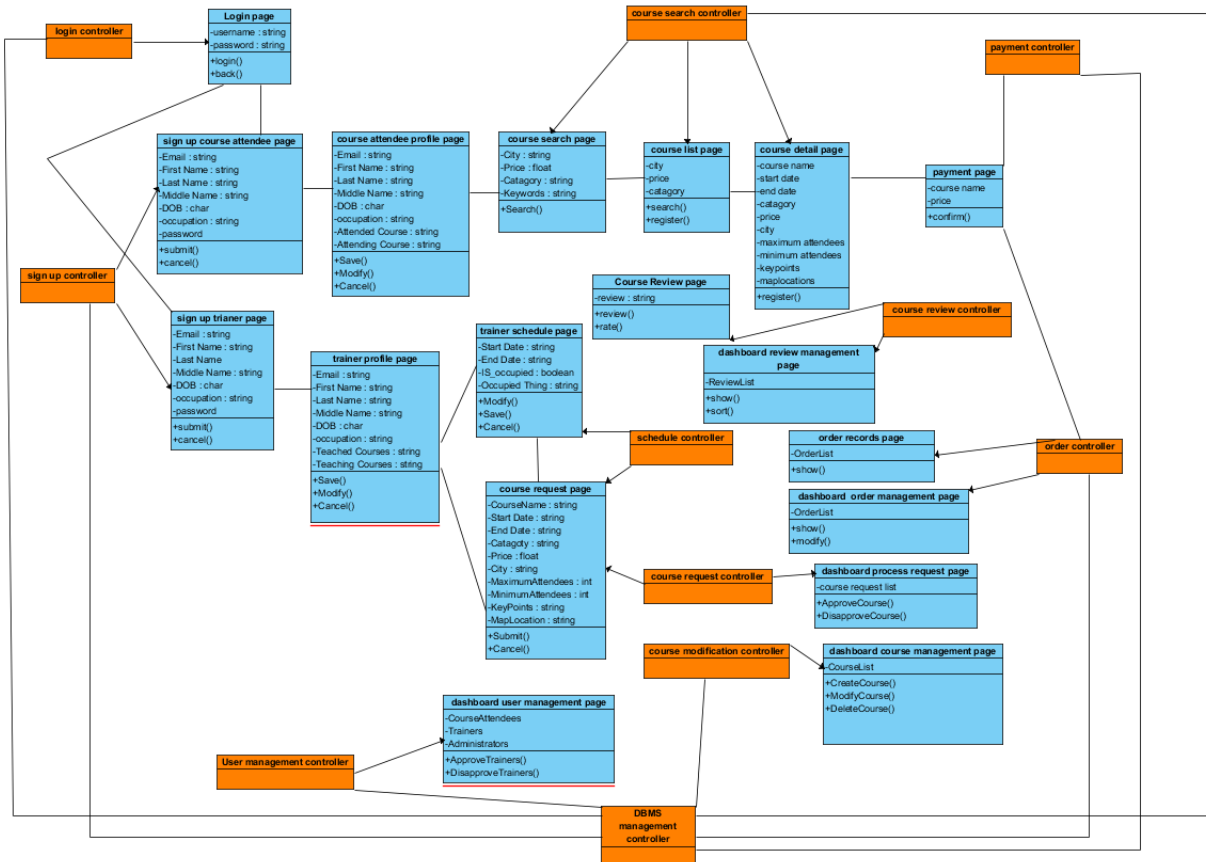


Figure 7: Design Class Diagram

Table 54: Design Class Description

Class	Type	Description
Login page	Boundary	Page with a login form
Sign up trainer page	Boundary	Page with a registration form of a trainer
Sign up course attendee page	Boundary	Page with a registration form of a course attendee
Course search page	Boundary	Home page with the function of searching
Course list page	Boundary	Shows the list of courses
Course details page	Boundary	Shows the detailed information of one course
Course review page	Boundary	Page with the function of review and rating
Order records page	Boundary	Shows the list of orders

payment page	Boundary	Page with a payment information
Course attendee profile page	Boundary	Shows the information of each course attendee
Trainer profile page	Boundary	Shows the information of each trainer
Trainer schedule page	Boundary	Page with trainer's schedule
Course request page	Boundary	Page with course request form that is filled in by trainer
Dashboard course management page	Boundary	Page with the function of course modify, create, delete, approve and disapprove
Dashboard user management page	Boundary	Page with the function of managing all users and trainers and approving or disapproving trainers' application
Dashboard order management page	Boundary	Shows the list of orders
Dashboard review management page	Boundary	Shows the list of reviews
Dashboard process course request page	Boundary	Page with the function of process the course application from trainers
Login controller	Control	Responsible for verifying login information
Signup controller	Control	Responsible for verifying signup information
Course search controller	Control	Responsible for searching courses according to specifications
Course review controller	Control	Responsible for users' reviews and ratings for courses and shows the summary of reviews and ratings for administrator
Course request controller	Control	Responsible for trainers submitting course request and administrator processing requests
Course modification controller	Control	Responsible for administrator modifying courses' information
User management controller	Control	Responsible for showing course attendees and trainers basic information and processing the identity application from trainers
Payment controller	Control	Responsible for the connection between the system and EventBrite
Order controller	Control	Responsible for recording generated orders and showing orders information for each user and administrator
Schedule controller	Control	Responsible for recording schedule information for each trainer; verifying correctness of course arrangement; and showing schedule list for administrator and users
DBMS controller	Control	Responsible for storing all the information of the website

3.1.3 Process Realization

3.1.3.1 search course

This is the use case of searching course which is the main function of our system. Users could search courses according to the filter or filling in keywords to complete search. If it failed then the whole system will be useless.

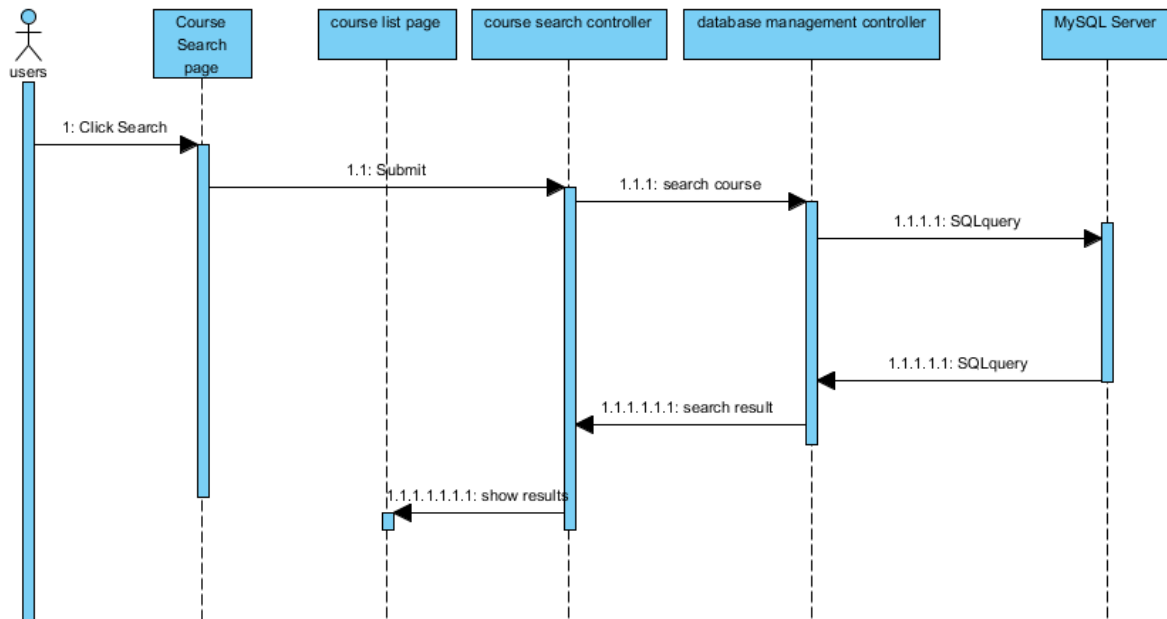


Figure 8: Process Realization Diagram—search course

3.1.3.2 register course

This use case comes with high risk because our system could be incompatible with the exterior payment system—EventBrite.

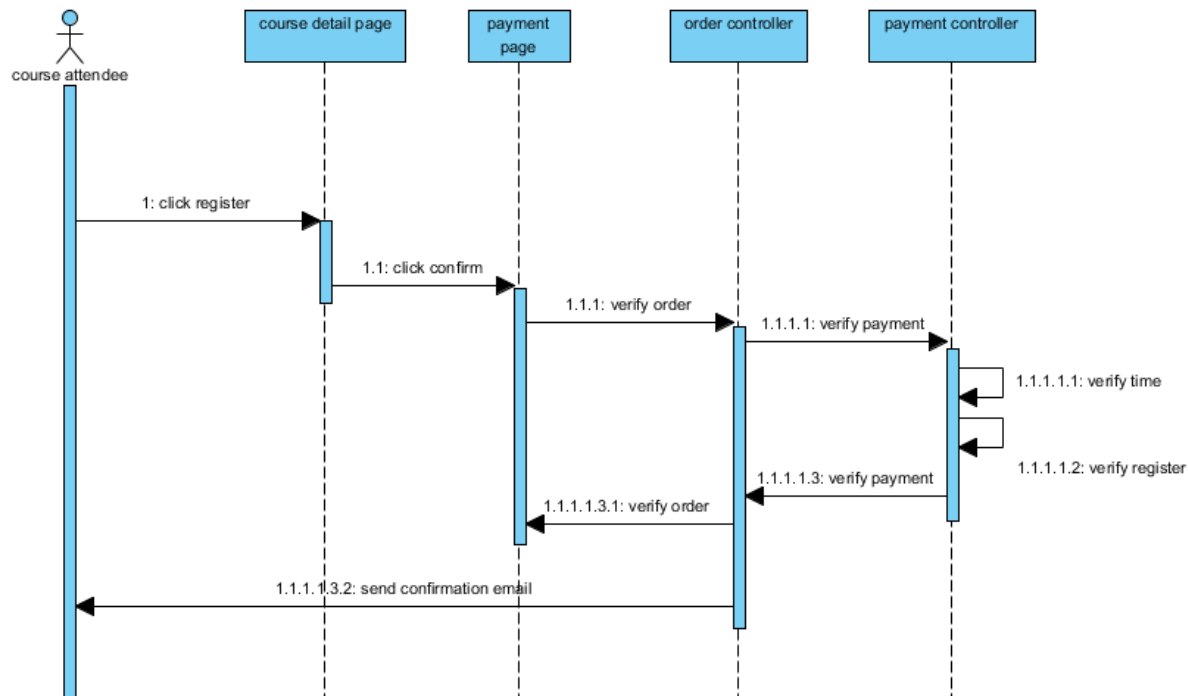


Figure 9: Process Realization Diagram—register course

3.2 Design Rationale

In our system, we use play framework to build the backend of the website. Play framework is a high-productivity Java and Scala web application framework that integrates the components and APIs we need for website development. Play framework is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption for highly-scalable applications thanks to its reactive model, based on Iteratee IO. That's why we choose this framework to development the backend of the website.

A Play application follows the MVC architectural pattern applied to the web architecture. This pattern splits the application into separate layers: the Presentation layer and the Model layer. The Presentation layer is further split into a View and a Controller layer.

The Model is the domain-specific representation of the information on which the website operates. Domain logic adds ‘meaning’ to raw data (e.g., calculating if today is the user’s birthday, or the totals and taxes for some selected courses).

The View renders the model into a form suitable for interactions, typically a user interface. Multiple views can exist for a single model, for different purposes. In our website the view is usually rendered in a ‘web format’ like HTML, XML or JSON.

The Controller responds to events (typically user actions) and processes them, and may also invoke changes on the model. In our website, events are typically HTTP requests: a Controller listens for HTTP requests, extracts relevant data from the ‘event’, such as query string parameters, request headers... and applies changes to the underlying model objects.

As for the payment function, we use the payment gateway—EventBrite to complete. It is the requirements from our client. The website we developed calls EventBrite API for course registration and course payment.

About the login part, users could use not only the account comes with sign-up function but also the account of Facebook. The website calls the Facebook API to create a way for users logging. What’s more, users in the website could share course information with friends. The share function also need Facebook API to complete.

4. Technology-Specific System Design

4.1 Design Overview

4.1.1 System Structure

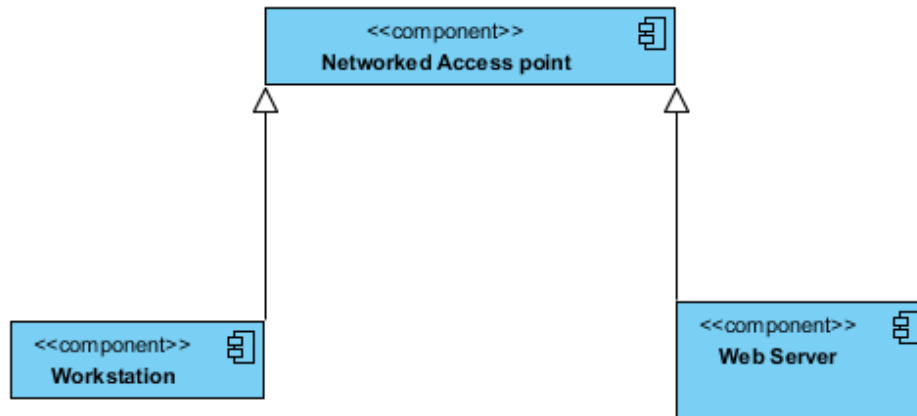


Figure 10: Hardware Component Class Diagram

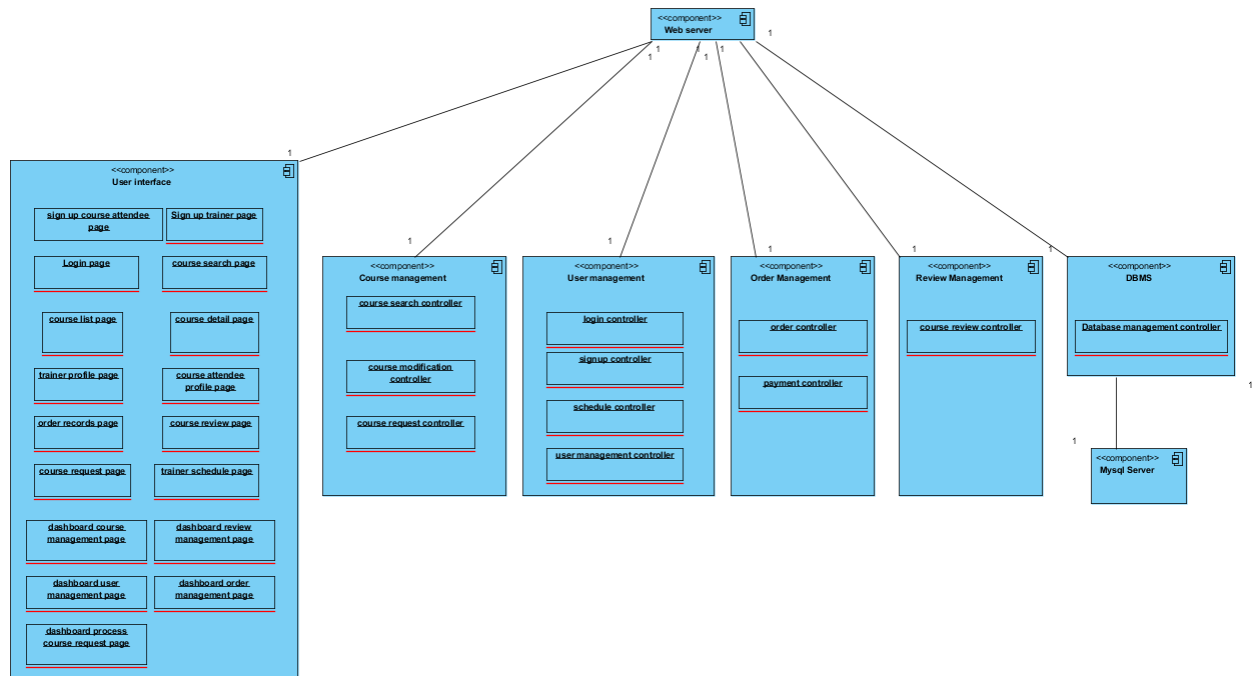


Figure 11: Software Component Class Diagram

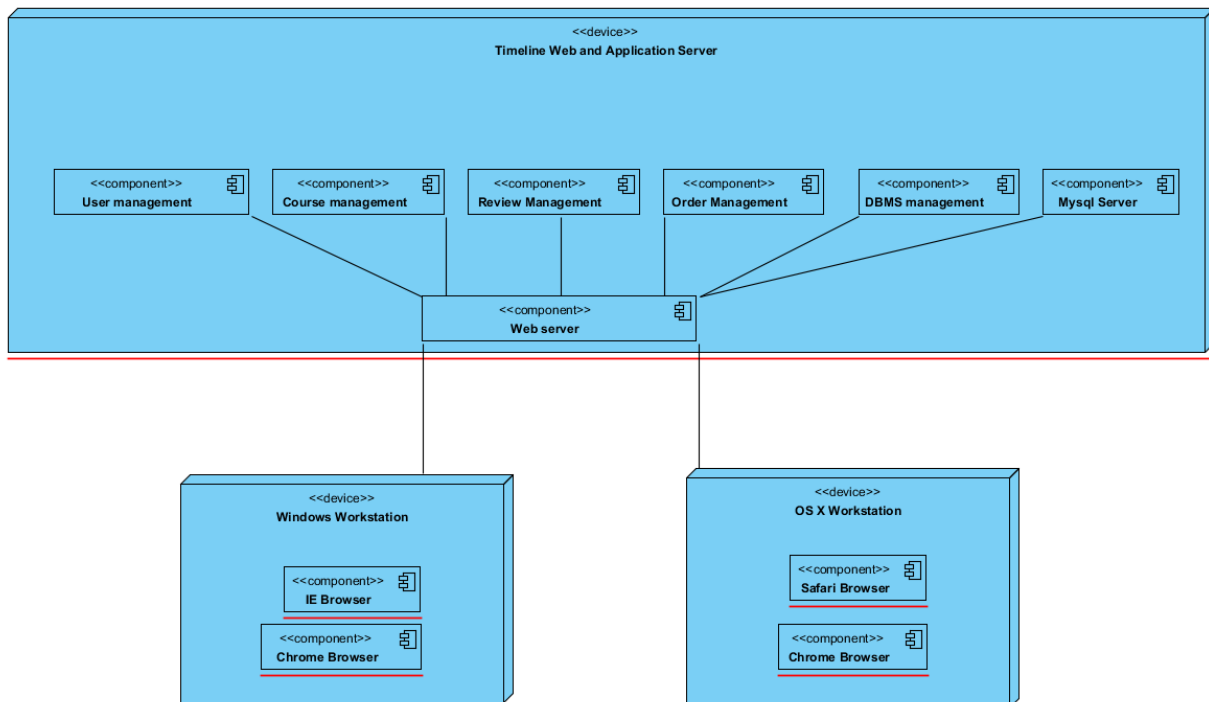


Figure 12: Deployment Diagram

Table 55: Hardware Component Description

Hardware Component	Description
Networked Computer	A computer that is connected to other networked computers through the internet. In our system, every computer will be networked in this manner.
Workstation	A networked computer which is used to access services on the internet. In our system, this will be computers used by people at home or in an office to interact with the ShareTheTraining
Web Server	A networked computer which provides applications to workstations. In our system, this will be the server our software will be deployed on.

Table 56: Software Component Description

Software Component	Description
User Interface Component	This component contains ShareTheTraining pages that users of the system interact with directly
Course Management Component	This component performs all the main functionalities of the course management process. These functionalities include: Approve/ Disapprove courses; Create/ Delete/ Modify courses.
User Management Component	This component performs functions involved in managing information relating to authenticated users of the ShareTheTraining System.
Order Management Component	This component performs functions involved in managing registration and payment of each order.
Review Management Component	This component performs functions involved in managing review and ratings.
DBMS	This is the Database Management System (DBMS) that stores all data used by the CSC Volunteer Tracking System.

Table 57: Supporting Software Component Description

Support Software Component	Description
My SQL	A relational database management system
Play Framework	An open source framework on which build the backend of website

4.1.2 Design Classes

4.1.2.1 User interface

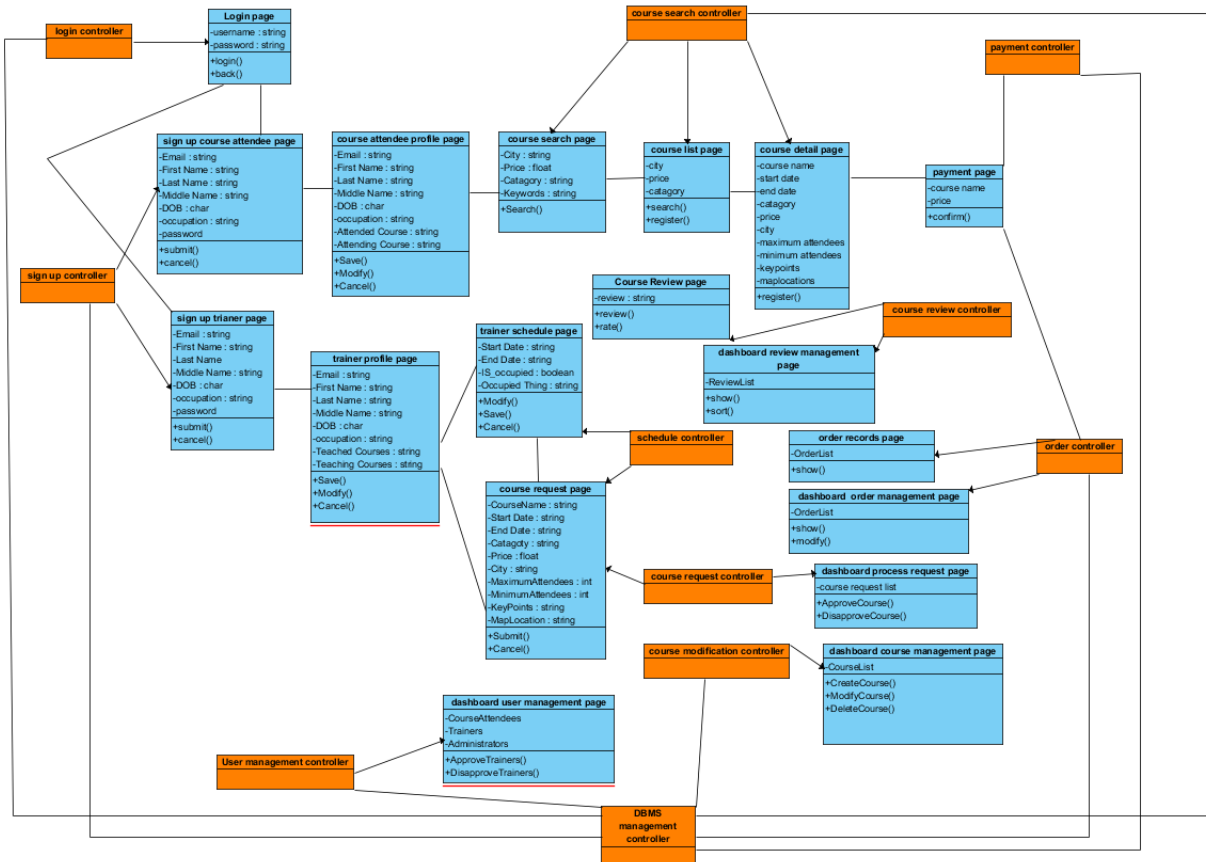


Figure 13: Design Class Diagram

Table 58: Design Class Description

Class	Type	Description
Login page	Boundary	Page with a login form
Sign up trainer page	Boundary	Page with a registration form of a trainer
Sign up course attendee page	Boundary	Page with a registration form of a course attendee
Course search page	Boundary	Home page with the function of searching
Course list page	Boundary	Shows the list of courses
Course details page	Boundary	Shows the detailed information of one course
Course review page	Boundary	Page with the function of review and rating
Order records page	Boundary	Shows the list of orders

payment page	Boundary	Page with a payment information
Course attendee profile page	Boundary	Shows the information of each course attendee
Trainer profile page	Boundary	Shows the information of each trainer
Trainer schedule page	Boundary	Page with trainer's schedule
Course request page	Boundary	Page with course request form that is filled in by trainer
Dashboard course management page	Boundary	Page with the function of course modify, create, delete, approve and disapprove
Dashboard user management page	Boundary	Page with the function of managing all users and trainers and approving or disapproving trainers' application
Dashboard order management page	Boundary	Shows the list of orders
Dashboard review management page	Boundary	Shows the list of reviews
Dashboard process course request page	Boundary	Page with the function of process the course application from trainers
Login controller	Control	Responsible for verifying login information
Signup controller	Control	Responsible for verifying signup information
Course search controller	Control	Responsible for searching courses according to specifications
Course review controller	Control	Responsible for users' reviews and ratings for courses and shows the summary of reviews and ratings for administrator
Course request controller	Control	Responsible for trainers submitting course request and administrator processing requests
Course modification controller	Control	Responsible for administrator modifying courses' information
User management controller	Control	Responsible for showing course attendees and trainers basic information and processing the identity application from trainers
Payment controller	Control	Responsible for the connection between the system and EventBrite
Order controller	Control	Responsible for recording generated orders and showing orders information for each user and administrator
Schedule controller	Control	Responsible for recording schedule information for each trainer; verifying correctness of course arrangement; and showing schedule list for administrator and users
DBMS controller	Control	Responsible for storing all the information of the website

4.1.3 Process Realization

4.1.3.1 search course

This is the use case of searching course which is the main function of our system. Users could search courses according to the filter or filling in keywords to complete search. If it failed then the whole system will be useless.

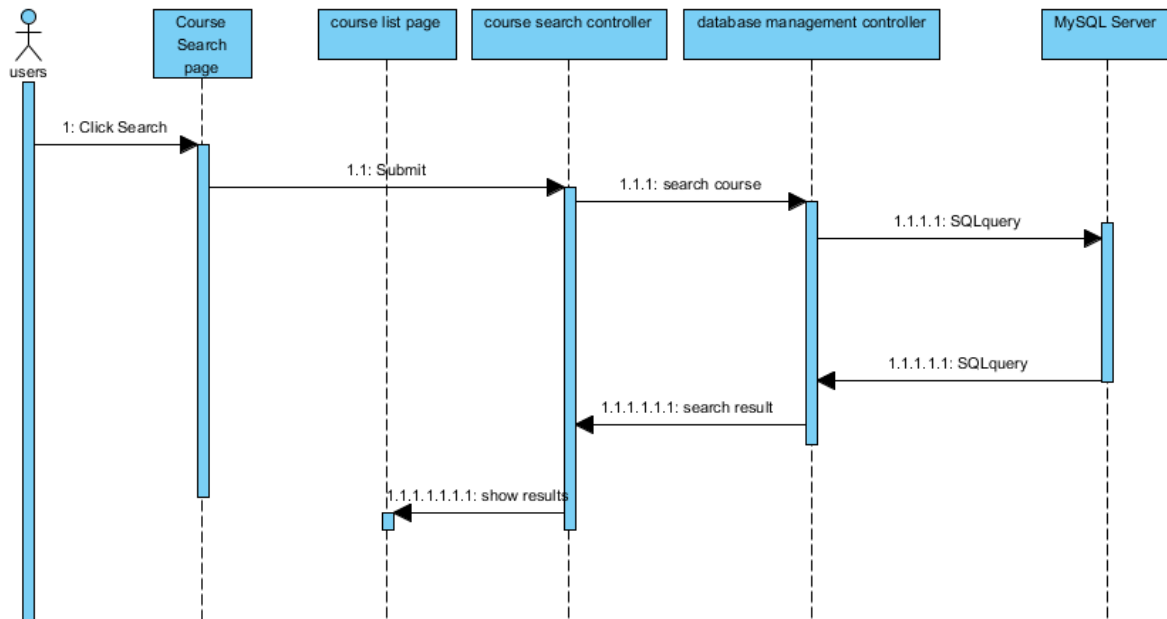


Figure 14: Process Realization Diagram—search course

4.1.3.2 register course

This use case comes with high risk because our system could be incompatible with the exterior payment system—EventBrite.

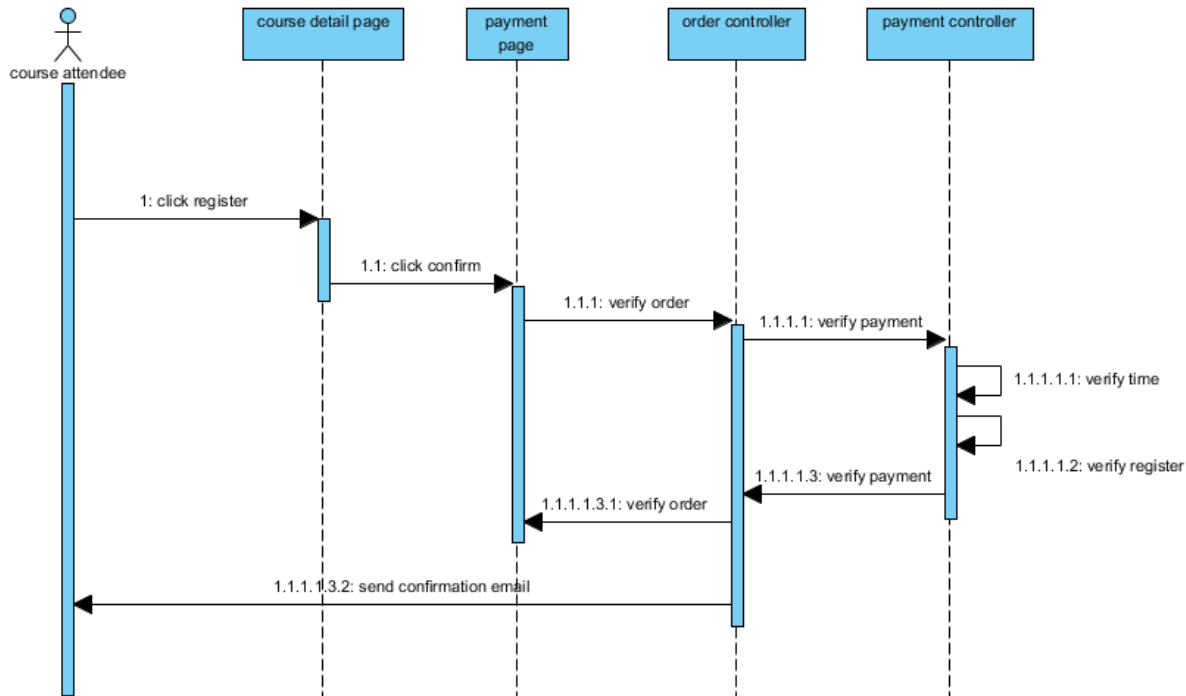


Figure 15: Process Realization Diagram—register course

4.2 Design Rationale

In our system, we use play framework to build the backend of the website. Play framework is a high-productivity Java and Scala web application framework that integrates the components and APIs we need for website development. Play framework is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption for highly-scalable applications thanks to its reactive model, based on Iteratee IO. That's why we choose this framework to development the backend of the website.

A Play application follows the MVC architectural pattern applied to the web architecture. This pattern splits the application into separate layers: the Presentation layer and the Model layer. The Presentation layer is further split into a View and a Controller layer.

The Model is the domain-specific representation of the information on which the website operates. Domain logic adds ‘meaning’ to raw data (e.g., calculating if today is the user’s birthday, or the totals and taxes for some selected courses).

The View renders the model into a form suitable for interactions, typically a user interface. Multiple views can exist for a single model, for different purposes. In our website the view is usually rendered in a ‘web format’ like HTML, XML or JSON.

The Controller responds to events (typically user actions) and processes them, and may also invoke changes on the model. In our website, events are typically HTTP requests: a Controller listens for HTTP requests, extracts relevant data from the ‘event’, such as query string parameters, request headers... and applies changes to the underlying model objects.

As for the payment function, we use the payment gateway—EventBrite to complete. It is the requirements from our client. The website we developed calls EventBrite API for course registration and course payment.

About the login part, users could use not only the account comes with sign-up function but also the account of Facebook. The website calls the Facebook API to create a way for users logging. What’s more, users in the website could share course information with friends. The share function also need Facebook API to complete.

5. Architectural Styles, Patterns and Frameworks

Table 59: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations
Play framework	An open source framework on which build the backend of website	Benefits <ul style="list-style-type: none">- Agile for developement- Free (open source)- Lightweight, stateless, web-friendly architecture Costs <ul style="list-style-type: none">- Free Limitations <ul style="list-style-type: none">- Lack of real success cases