COMP 3211 Software Engineering

Software Requirements Specification

24 November, 2017

eBoard

<E-learning Course Application Website>

Project members

Tsui Kin Min (15076132D)

Lam Tsun Fung (15067703D)

Ng Tsz Kin (15066736D)

Contents

1. Introduction

1.1 Goals and objectives

eBoard is a course registration website aims to provide online platform for people to learn practical knowledge and skills. Course registration is available in eBoard for registered users of eBoard which provide economic income to the company. The potential users could be students and adults for academic or advanced studies. eBoard aims to provide non-traditional curricular schooling which users can register their favorite courses for learning with available session. It can be morning, afternoon and evening classes in weekdays and weekend. Great variety of courses on eBoard allows users to find the one in need. Course information and registered details can be found on eBoard by a simple click conveniently, such as course schedule, venue and information of the teacher. A customized learning experience is provided by eBoard.

1.2 Statement of scope

eBoard is a web-based course registration platform aiming to provide useful courses for teenagers and adults on learning purpose. Users can reduce the time spending on searching desired courses through eBoard. Course information and user personal data will be stored in database for better management. Furthermore, SQL is used to provide efficient information seeking and website content display.

The major input would be users' input. Especially, users input their username and passwords for user authentication. SQL will be processed in the background for accessing the records in database. After the identity authentication, searching criterias which are data, time and venue of courses will be input for users in advanced filtering and searching to target courses. Once eBoard receives users' input, system will compare users' input username and passwords with the record of users' account stored in the database. The match case of comparison means successful user authentication. The returned output would be the notification of successful login, user personal profile and required searching results.

Essential requirement

- User authentication :
  A successful comparison between users' inputs and database records will authenticate users' identity which a visitor already register an account on eBoard..
- User authorization :
  Only certain role of user can access certain authorized pages or otherwise their action will be recorded down and redirected. In detail, user shouldn't have the permission to alter their own as well as others' permission level while admin also shouldn't have the permission to alter the user profile of any user.
- Secured data transfer :
  All of the HTTP Request and Session data should be encrypted or adopt a symmetric key exchange authentication scheme to ensure the data integrity and also the connection between server and the client.
- Keywords searching :
  By entering keywords related to the courses, visitors and users are able to find out the courses they want in the search bar.
- Course registration :
  Registered users are able to click "register" button to register a course for learning on eBoard. Registration records will be stored in the database.
- Payment handling :
  eBoard should allow users to select the payment method and handle the money transfer in third-party manner by cooperating with credit card companies.
- Registration history :
  The users can check the information for the registered courses such as schedule, venue.

Desirable requirement

- Advanced filtering :
  Using further filtering tools like checkbox and drop-down menu, visitors and users are able to search for more accurate courses they looking for.
- News publication :
  Some news and notifications which can be course discounts or course dalayment will be shown on the homepage or directly send as user personal notifications in order to inform users for important news.
- Upload and download teaching materials :
  Teachers can create folders and upload required teaching contents to the specific course page and students can download the contents.

- Add to favourite
  Allow users to add specific course to their own favourite list for future reference.

Future requirement

- Course recommendation :
  eBoard aims to recommend suitable course to users automatically by gathering and making analysis on users' personal information. Thus, users may look for courses perfectly satisfied.
- Comment system :
  Users are able to comment on courses which give suggestion to other users whether or not they should register the courses.
- Discussion board :
  Registered students within the same course and the teacher can be discuss with each other and seek help from others which enhance learning proficiency and efficiency.

1.3 Software context

eBoard is a website which provides an e-platform for both teenagers or adults to find and register courses they want for education purpose. Visitors and users can both search courses and check their details to seek for suitable lessons. To register any courses, visitors need to create and sign in an account before. Registration history will be record inside the database and it helps eBoard to manage users' activities. Courses charge users money which users pay the fee in online by credit card conveniently. eBoard provides a user-friendly interface that users can easily follow the course details. These basic functions should be existed in the early stage of eBoard making registration possible. In the future, more accurate searching result and personalized course suggestion will be made to users. More improvements and upgrades will be done on eBoard and it may increase the number of course registration.

1.4 Major constraints

Several constraints may be occurred during requirement specification state. It would be the different opinions on software requirements from stakeholder. Stakeholders and customers may not know and how to express the exact needs they want. It makes requirement specification become harder and slower. Deciding and confirming the exact software requirements to be satisfied is difficult and time-consuming. The software requirements affect the development and design of eBoard mostly. Laws, norms, guidelines and style may impact on the function design of the eBoard. There is low probability to have recruitment on experienced staffs. New technical team members may not be familiarized with eBoard. It affects the development progress seriously.

2. Usage scenario

This section provides a usage scenario for the software. It organizes information collected during requirements elicitation into use-cases.

2.1 User profiles

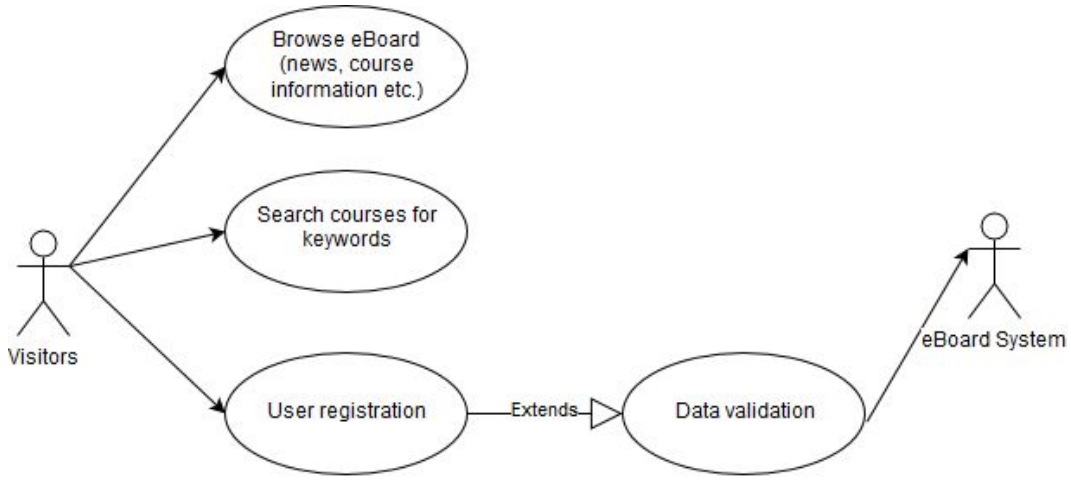
After the identity authentication, eBoard advancely authorize users' privileges according to the user types. Users of eBoard can be clearly distinguished in 3 different types which are visitors, users and admin. Normal visitors can register an account in eBoard and become a user having higher privileged and function. Only higher staffs and engineers of eBoard can login to eBoard as admin with the highest authority,

User Type	Visitor	User (student)	User(Teacher)	Admin
Previledges	● Browse the website ● Search for courses	● **Do whatever visitors can** ● register courses ● Pay for courses ● Add the selected courses to favourite list ● Leave comments to courses ● Access to the course homepage	● **Do whatever users can** ● Upload teaching material ● Talk and manage the discuss board ● Read only to simple information of students	● **Do whatever users can (for testing purpose)** ● **manipulate** the database ● monitor and edit the website ● announce news

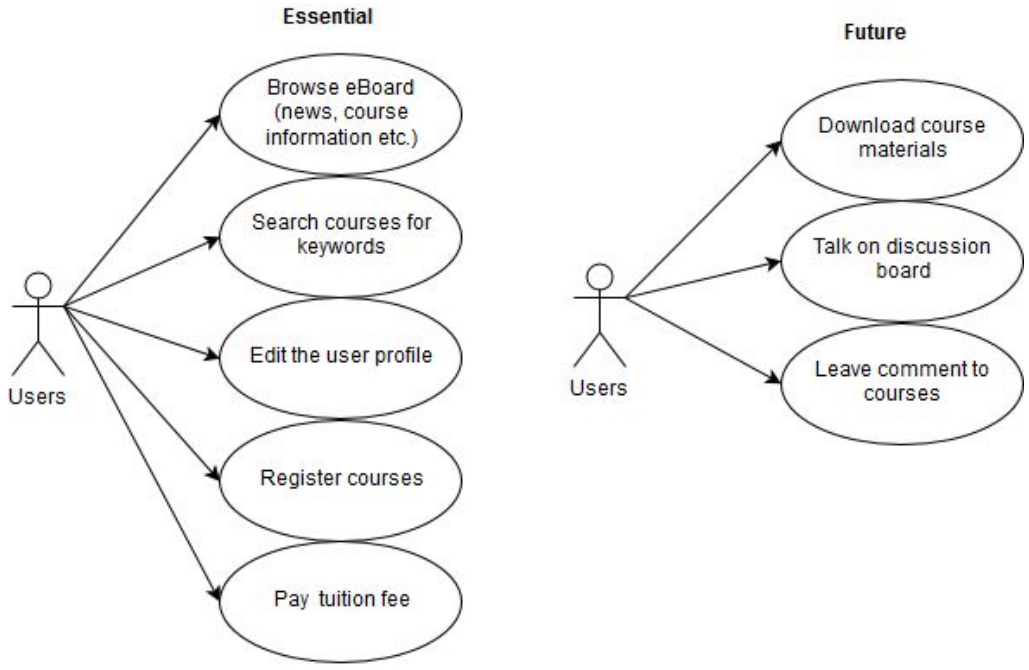
2.2 Use-cases

2.2.1 Use-Case Diagram

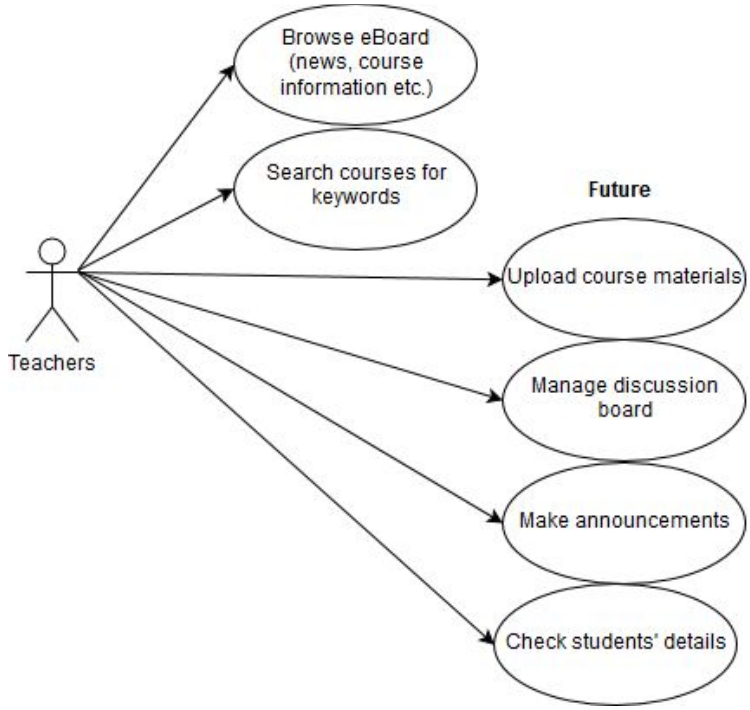
Visitor's use-case :



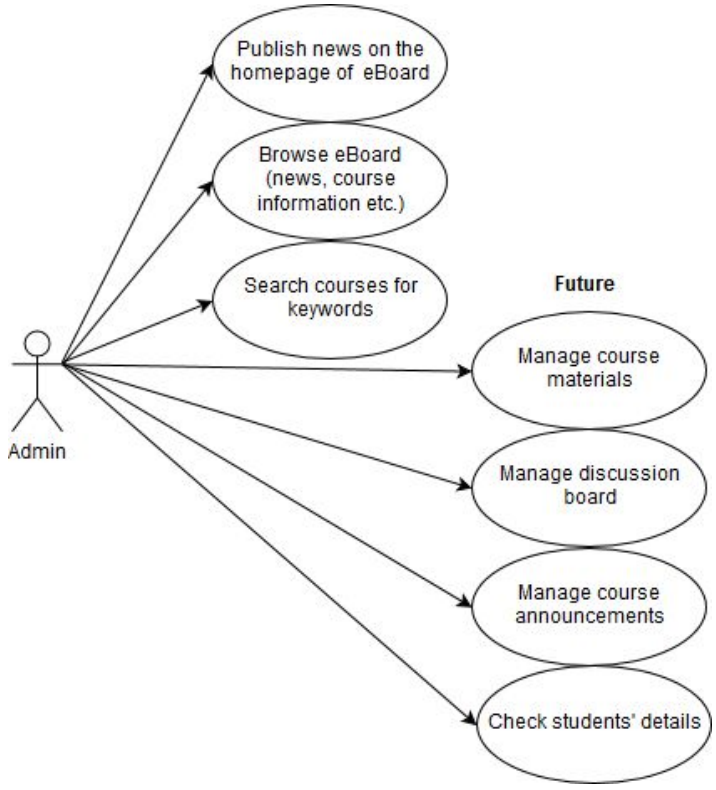
User's use-case :



Teacher's use-case :



Admin's use-case :



2.2.2 Use-Case Descriptions

Admin has the highest privilege on eBoard which he/she does not register courses at all but try the functions mainly for testing, fixing and upgrade.

Student is a kind of the user types who can download course materials and use the discussion board.

Teacher is another kind of the user types but they own special features such as uploading course materials and manage the discussion board in the future.

Usage	Description
Browsing on eBoard	Visitors, users can access to eBoard by URL and browse the website such as common news and course details.
Searching	Visitors and users can search the course by entering keywords or advanced filtering function in the search bar.
Course Details	Visitors and users can find the details of courses such as course schedule and venue through some hyperlinks.
Account Registration	Visitors can register an account to become user for more functions by entering valid personal data on registration page.
Payment	Users can choose the payment methods and pay the tuition fee for the selected courses.
User Profile	Users can access its user profile checking their own personal information, especially the favourite list.
Upload Teaching Material	Courses related teachers can upload and manage associated learning contents to the course homepage.
Download Teaching Contents	Students can download the course materials from the course homepage.
Discussion board	Students can talk on discussion board while teachers can manage the board.

2.3 Special usage considerations

In special scenario, followings function will be processed.

Usage	Description
News Publication	Admin can announce important news towards the homepage of eBoard for notification.
Manage Course Announcement	Teachers can create and edit the course announcement to the course homepage. (admin seldom do that)
Manage Discussion Board	Teachers and admins can monitor or delete the conversation on the board.
Check Students' Basic Information	Teachers and admins can look for students' simple data for educational purpose only.

3. Data Model and Description

This section describes the information domain for the software.

3.1 Data objects

Admission:

Attribute	Description
Admission ID	An unique ID assigned to each admission done by users (PK)
User ID	An ID number used to represent the user who made an admission (FK)
Course ID	An ID number used to represent the course associated to the admission (FK)
Creation Time	The time of the admission being recognised and recorded down (TIMESTAMP)

Blacklist:

Attribute	Description
Blacklist ID	An unique ID assigned to each authorized access being recorded (PK)
User ID	An ID number used to represent the user who attempt an unauthorized action (FK) / which can be a null number at the same time since non-user may have such attempt also (NULLABLE)
IP Address	An IPv4 IP address used to represent the device's digital address (Private IPv4 Address)
Creation Time	The time of the new blacklist record being recognised and recorded down (TIMESTAMP)

Comment:

Comment ID	An unique ID assigned to each newly added comment by user (PK)
Event ID	An ID number used to represent the ownership of each event towards the comment (FK)
User ID	An ID number used to represent the user who posted the comment (FK)
Comment	The content of the comment itself (TEXT)
Creation Time	The time of the new comment being recognized and recorded down (TIMESTAMP)

Course:

Course ID	An unique ID assigned to each newly added course (PK)
Course Name	The name of the course (STRING)
Course Duration	The duration of the course (STRING)
Course Level	The level of the course (STRING)
Course Location	The location of the course (STRING)
Creation Time	The time of the new course being recognized and recorded down. (TIMESTAMP)

CourseDetail:

CourseDetail ID	An unique ID assigned to each pivot record (PK)
Course ID	An ID number used to represent the course (FK)
Discipline ID	An ID number used to represent the discipline associated to the course (FK)
User ID	An ID number used to represent the user who register the course (FK)

Discipline:

Discipline ID	An unique ID assigned to each discipline (PK)
Discipline	The name of the discipline (STRING)
Creation Time	The time of the new discipline being recognized and recorded down. (TIMESTAMP)

Event:

Event ID	An unique ID assigned to each newly added event (PK)
User ID	An ID number used to represent the organizer of the event who is usually the Speaker of the platform(FK)
Event Speaker	Containing the name of invited guest separated by “,” (STRING)
Event Location	The location of the event (STRING)
Event Venue	The venue of the event (STRING)
Event Title	The title of the event (STRING)
Event Content	The main content of the event describing the flow and other details (TEXT)
Creation Time	The time of the new event being recognized and recorded down. (TIMESTAMP)

Message:

Message ID	An unique ID assigned to each newly added message by visitor (PK)
Guest Name	The name of the guest (STRING)
Email	The email of the guest (STRING)
Phone	The phone of the guest (STRING)
Message	The main message of the guest want to deliver and ask towards the platform (TEXT)
Creation Time	The time of the new message being recognized and recorded down. (TIMESTAMP)

Notification:

Notification ID	An unique ID assigned to each newly added notification triggered by the action of the admin (PK)
Notification Type	The type of notification (STRING)
Notifiable	Containing both the name of the model which issue the notification and a randomly generated id number for verification (STRING,ID)
Data	Containing a JSON-Like data structure which will store the detail of the notification (TEXT)
Read Time	The time of the notification being read (TIMESTAMP)
Creation Time	The time of the notification being recognized and recorded down (TIMESTAMP)

Permission:

Permission ID	An unique ID used to identify different permission assigned to each user/admin (PK)

Permission name	The name of the permission (STRING)
Guard name	The name of the guard which is a collection of middleware used to determine if certain action/route/page is authorized (STRING)
Creation Time	The time of the new permission being recognized and recorded down. (TIMESTAMP)

Post:

Post ID	An unique id assigned to each newly added post (PK)
Post title	The title of the Post (STRING)
Post body	The main content of the post (TEXT)
Creation Time	The time of the new post being recognized and recorded down. (TIMESTAMP)

Role:

Role ID	An unique id used to identify different role of the user (PK)
Role name	The name of the role (STRING)
Guard name	The name of the guard which is used is a collection of middleware used to determine if certain user has specified role (STRING)
Creation Time	The time of the new role being recognized and recorded down. (TIMESTAMP)

User:

User ID	An unique id assigned to each newly registered user (PK)
User Name	The name of the user which will be used for official record of post/course/event attempt. (STRING)
User Email	The email of the user which will be used as the login credential of the platform. (STRING)
User Password	The password of the user which will be used as the login

	credential of the platform which is a hash value instead of simple plaintext. (STRING)
Remember Token	A random generated token used to authenticate the “Remember Me” feature (STRING)
Stripe ID	A random generated ID number used for Stripe to identify the customer to be valid request (FK)
Card Brand	The brand of the card used by the user to perform purchase. (STRING)
Card Last Four	The last four digit of the credit/debit card (STRING)
Trial End Time	The end time of the trial given to user to try the feature of the platform (DATETIME)
Creation Time	The time of the new user being recognized and recorded down. (TIMESTAMP)

User Profile:

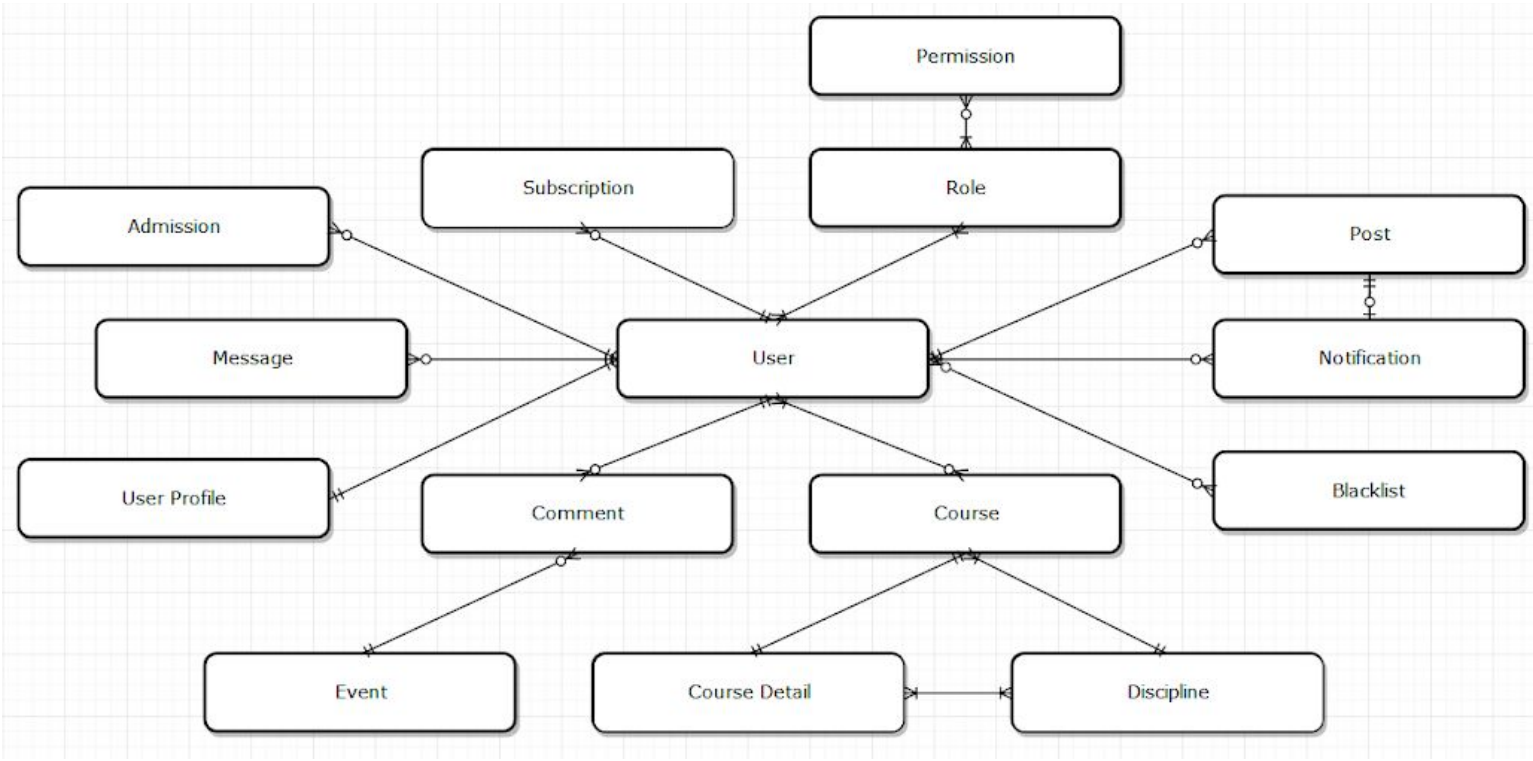
User Profile ID	An unique ID assigned to each newly added user profile (PK)
User ID	An unique ID used to reference the user who owns the user profile (FK)
User nickname	The nickname of the user which will be used to identify the owner of comment related to event/course (STRING)
User age	The age of the user (INTEGER)
User gender	The gender of the user (STRING)
User DOB	The date of birth of the user (DATE)
User mobile	The mobile number of the user (STRING)
User country	The country of the user (STRING)

Subscription:

Subscription ID	An unique ID assigned to each newly added subscription (PK)

User ID	An ID number used to reference the user who made the subscription (FK)
Subscription name	The name of the subscription (STRING)
Stripe ID	A randomly generated ID number for Stripe to authenticate whether the subscription is valid or not (STRING)
Stripe Plan	The name of the plan of Stripe specified by the subscription (STRING)
Quantity	The number of subscription issued by the user (INTEGER)
Trial End Time	The end time of the trial of the subscription (TIMESTAMP)
End Time	The end time of the subscription (TIMESTAMP)
Creation Time	The creation time of new subscription being recognized and record down (TIMESTAMP)

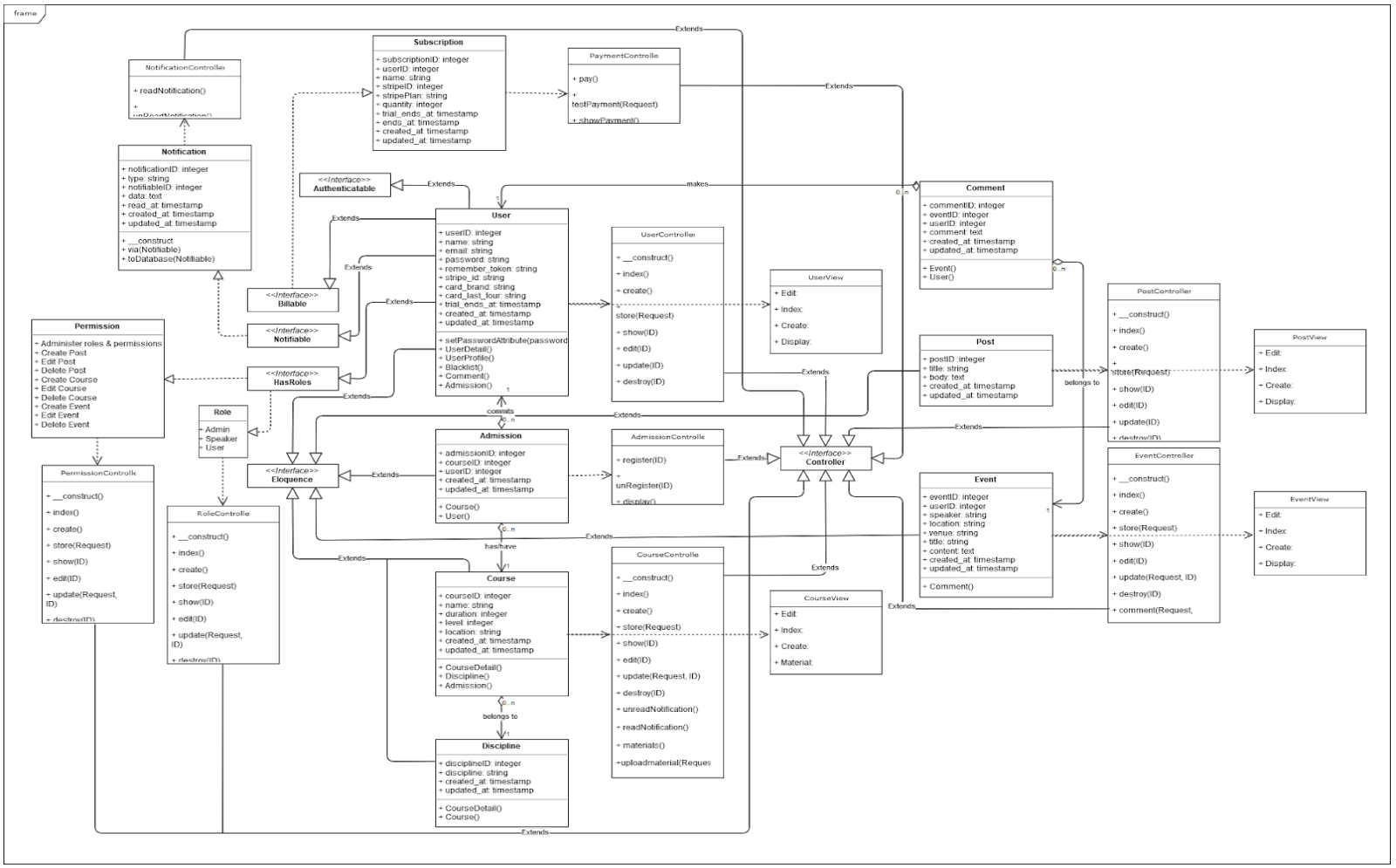
3.2 Relationships



3.3 Complete data model

4. Functional Model and Description

4.1 Class diagrams



4.2 Software Interface Description

Interface of the eBoard is mainly using the object-oriented decomposition. Heritance can be observed clearly from the class diagram. Heritance allows attributes and functions of components to be able for the purpose of reuse. It saves time and ease the difficulty of the system interface. Both normal users and engineers can understand the relationship between components easily under an OO interface. The difference is that user figure out it by an user-friendly interface without knowing the work behind codes but just the experience. To cite an example, users understand that

they need to login as an user so as to edit and view user related profile. In the background of the function, OO relationship simplifies the dataflow of the system and gives hints on future architectural design.

Simple Highlighted Example :

User Profile - - (inherit from) - -> User

User Registration
- Visitor A (doing user registration process to become User A)
Interface of Visitor allows visitor A to do user registration. Visitor A input registration data to the interface of User Registration. Data transfer to User interface to check existence of input after data validation. If valid and non-duplicated inputs are made, the registration is successfully.

User Profile
- UserA (doing user registration process)
The interface of User Profile is inherit from interface of User by selecting parts of the data to be displayed (e.g. nickname, email, gender). Thus, the displayed data actually is retrieved from another interface without generating new data. That is User A only can access User A related information.

Delete User(Authentication) - - (Check) - -> Permission - - (inherit from) - -> User

Delete User (e.g as an Admin)
-Alex (Logged in as Admin, where Admin can delete user due to the permission_id stated in Permission interface, and the related user_id in Permission interface is got from User interface)
Permission interface is inherit from User interface for getting the user id for storing related permission id. Delete User interface will access Permission interface and check whether Alex' user_id are related to the permission_id of admin which owned the right to delete user.

4.2.1 Internal system interfaces

As stated as the above which OO Interface is used to shape the system. Components of the system is treated as an object. Relationships used to link objects together and describe its behaviours. Since website is builded on database and provide information by accessing the database, the interface of website system can be basically divided according to roles in database. Common roles are users and admin which distinguish the authority of participants of the system.

An interface of a website is more like a view. eBoard system has many interfaces related specific function such as events, posts, user or course registration. In an abstract way, eBoard has 4 interfaces to manipulate on accessing other interfaces. They are :

-Visitor
-User
- Speaker (behaves like “Teacher”)
- Admin
These interfaces decide the access of interface. For example, visitors cannot access the interface of course registration unless they are registered and logged in as an user. Admin can easily access to most of the attributes and functions on different interfaces. Thus, the interface design gives direction on designing the architecture.

Database management system is another system entry for programmers to access and manage the database such as MySQL without login as admin in the website. However, system users still need to authenticated as admin before accessing the DBMS. There are graphical and command line user interface. The interface now is based on the entity relationship of attributes and tables. For instance, HKPOYU teachers need to choose the database named *HKPOLYU* before accessing the exam results of students.

4.2.3 Human interface

Users enter eBoard will be redirect to a clear and well distributed homepage including information bar in the middle, control panel at the top right hand side corner. Well known website element will be include to provide an user-friendly interface such as textfield and drop-down menu.

Admin interface has the major difference to normal user. Admin will be redirected to another interface once admin account is used. Admin can use powerful function to monitor the website. For example, admin can check registration record made by users and registration number each day. A special feature uniquely for admin is that payment record and detail(time, amount) are able to be viewed in admin panel. Hence, admin can monitor user actions to ensure the availability of daily services.

Mobile interface is mainly different to the computer one. Since the screen size of mobile phone is always small and varied a lot, different mobile display interfaces are required to be further adjusted and designed. Common screen size may fall into the range of 4.7’’ to 5.7’’ which should be considered first and tablet size also. eBoard will reduce its original size of interface according to the ratio difference between them.

The interface should be user-friendly which reduces the time of visitors and users spending on accessing the website, searching for specific courses and finishing course registration.

- Big enough font size : show the contents of eBoard clearly.

- Clear distributed interface : well-organised for efficient uses (search bar, drop down menu etc.).
- Using soft colors : comfortable for users to view the website.

5. Behavioural Model and Description

5.1 Description for software behaviour

5.1.1 Events

User:

1. register()
2. addUser()
3. editUser()
4. destroyUser()
5. searchUser()
6. showUser()

Course

1. addCourse()
2. editCourse()
3. destroyCourse()
4. searchCourse()
5. showCourse()
6. registerCourse()
7. unregisterCourse()
8. pay()

Portal

1. Login()
2. Logout()

Event

1. addEvent()
2. editEvent()
3. destroyEvent()
4. showEvent()
5. addComment()
6. editComment()

Notification

1. showNotification()
2. readNotification()
3. addNotification()

Role/Permission

1. addRole/Permission()
2. editRole/Permission()
3. destroyRole/Permission()

5.1.2 States

Login

The state of the platform will turn from offline to online after user choose to register or login to the platform

User

The state of the platform will turn to “User” related mode that Create, Retrieve, Update and Delete can be performed on the model with the same namespace, i.e. User.

Course

The state of the platform will turn to “Course” related mode that Create, Retrieve, Update and Delete can be performed on the model with the same namespace, i.e. Course.

Event

The state of the platform will turn to “Event” related mode that Create, Retrieve, Update and Delete can be performed on the model with the same namespace, i.e. Event.

Notification

The state of the platform will turn to “Notification” related mode. Users can read notification while Admin or Speaker can create notification. The model with the same namespace, i.e. Notification, will be modified.

Role/Permission

The state of the platform will turn to “Role/Permission” related mode that Create, Retrieve, Update and Delete can be performed on the model with the same namespace, i.e. Role/Permission.

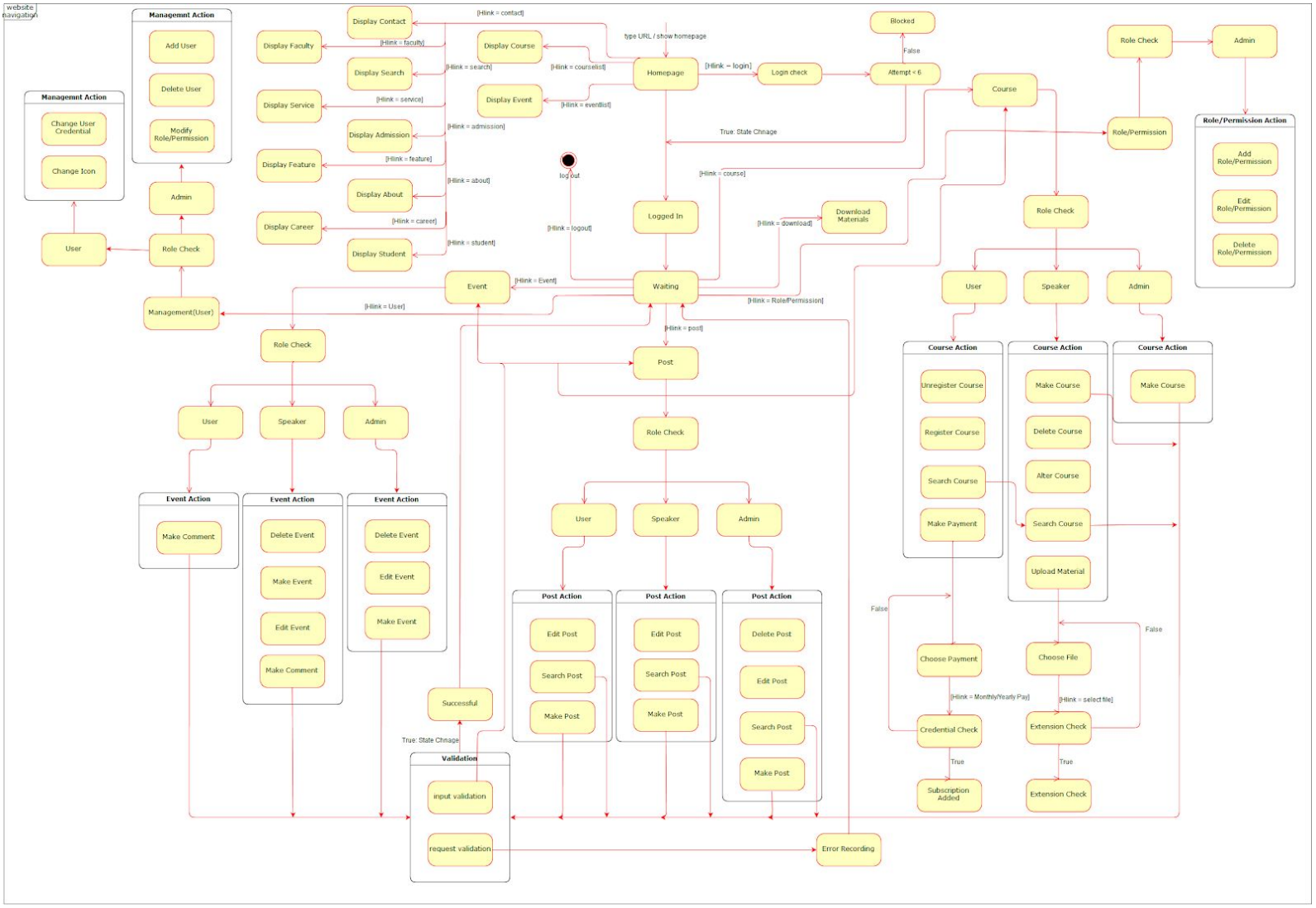
Guest

The state of the platform is at “Guest” mode by default that every visitors/users can have basic exploration to the platform such as navigation to admission, contact, service and faculty pages.

Logout

The state of the platform will turn from online to offline after user choose to logout the platform

5.2 Statechart Diagram



6. Restrictions, Limitations, and Constraints

Conflict of requirements

The website should be developed in several stages mainly using waterfall model version by version. However, the stakeholders of business may have different points of view on the website. For instances, they may argue on how to display the course list and what criterias to sort the course. Conflict on requirement specification stage may lower the development progress. A meeting is essential for a correct and efficient development. A report of system requirements can also illustrate the goals during the development.

Budget and schedule

During the development stage, the design and testing progress may not follow the schedule due to the technical issues. To cite an example, bugs may occur in the codes and it leads error. Extra time is needed to do fault location and fixing. As a result, over budget may be happened. To prevent these problems, software development may not be on time. Well attention and communication is required among technical teams to avoid bugs. Establishing a timetable and plan

System representation

eBoard is a completely new website which may consume much time for users to figure out all the functions. A clear interface design should be made to the website. To help users on faster manipulation of the website, a user guide is essential to nurture users about the information of eBoard.

Unstable service

Bugs associated code may cause system to be crashed and terminate the website services. A special team should handle related problems on time to provide sustainable services. New technical team members may be difficult to understand the system. A system manual and training are important to familiarise eBoard faster.

7. Validation Criteria

7.1 Classes of tests

In order to confirm the software requirements whether it fulfills the needs consistently or not, validation is needed to test the website functions. Instead of testing the system based on the code using white-box testing, a simple black-box testing will be used. Texts will be mainly typed as input for testing. By observation the output, technical team can then validate the requirements.

User Registration
-username, password, email address, nickname and country etc.
Username, password and email address should be validated and they should be unique except the password. Checking the existence of input username and email address inside the database can ensure the uniqueness of these fields. Format check is applied to email address and password whether they are consist of '@' and 'at least one upper and lower case' respectively. Moreover, the password should be at least 6 letter long. Data verification is worked on password field which double inputting the password helps visitors to check the typing mistake. The system will check on the data completeness for validation too.
Test case:

1. Username : peter (already exist in the database)
2. Password :Abc123 ; Confirm Password : Abv123
3. Email address : Abc123@gmail.com (unused email)

User Login and Logout

Only visitors input the correct username and password can successfully login the account as users, Users can easily logout their account by just clicking the button at the top right hand side corner,
Test case:
1.Username : peterlam ; password : Peter123 (account existed but wrong password)
2. Username : Admin123nimdA ; password : AdminnimdA (account existed and correct username and password)
3. Click the logout button and see whether login status still there or not

Change Password

-original password, new password, confirm new password
The new password is required to be 6 letter long, at least one upper and lower case letter. Confirm new password should match with the password perfectly However, the original password should be correct so as to change the password successfully.
Test case:
1. Original password : Abd123 (not same as the original one)
2. New password : Def123 ; Confirm new password : Def123 (correct original password)

Search Bar and Post Search

-dropdown menu, keyword search
The options of dropdown menu should be all valid. Related courses should be shown if their names are matched with the input keyword.
Test case:
1. Select "PHP" as discipline, “1” as level (drop down menu)
2. Type in “Top 10 practical course“ (keyword search)

Course Registration and Payment

Check system whether users can register courses and pay the fee or not.
Test case:
1. Simply try registration and payment as a visitor
2. Simply try registration and payment as a user

Post and Event Establishment

Users can post posts to share opinions by entering the topic and content. Admin can further publish events by inputting the title, tutor name, venue, time and content.
Test case:

1. Users (JohnLegend) establish a post and check it in the post page
2. Admin build an event on a specific day and find it in the event page (Welcome day; tutor are Paul, Peter and Mary)

7.2 Expected software response

User Registration

1. Fail, notify the user by "username is taken already, please choose another username"
2. Fail, the confirm password is wrong
3. Success, a new email and suit the email format

User Login and Logout

1.Fail to login, the password is wrong
2. Successful login, username and password is correct
3. Successful logout, simple observation which nickname of the user and the user profile button are no longer exist in the top right hand side corner

Change Password

1. Fail, wrong original password
2. Success, original password correct and new password match the confirm new password

Search Bar and Post Search

1.The level 1 PHP courses will be displayed to the screen
2. The post which its title contains the keyword will be displayed to the screen

Course Registration and Payment

1. The register button is in grey color and unclickable to visitor
2. User can click the register button and see the registered results and payment at his/her own user profile.

Post and Event Establishment

1. The post written by JohnLegend can be found
2. The even named "Welcome day" having Paul, Peter and Mary can be found on event page

Once these test case can be performed in above manners, software requirements are consistently satisfied.

7.3 Performance bounds

Response Time

Since user hate variation response time, they may do something else while just waiting there. Long waiting time will drop users' attention. 0.1 second is the ideal time that users feel the system being

responded to themselves instantaneously. The response time should be well defined unambiguously and concisely.

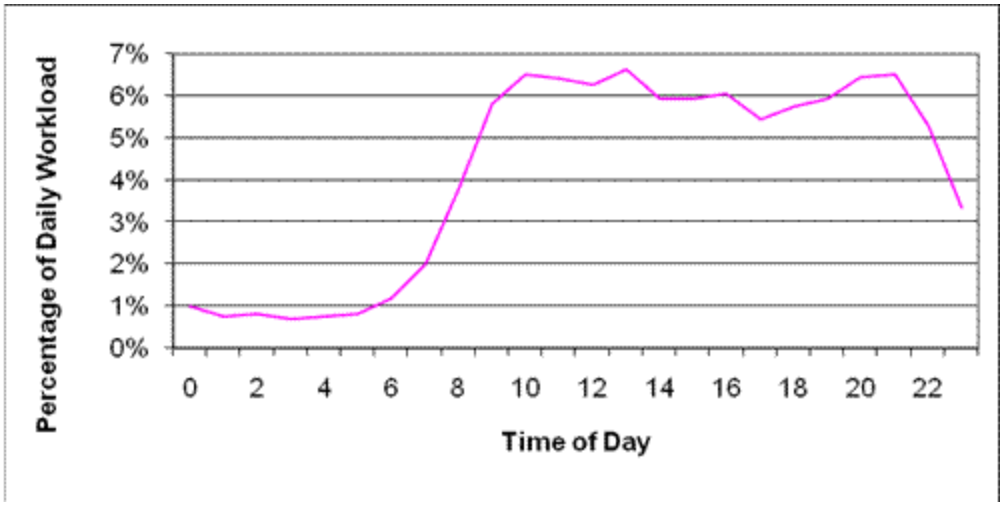
Page	Time
Search, Post, Course List, Event	a fixed 1 s in normal at most 5 s
Login Page (avoid online brute force attack on users' accounts.)	a fixed 5 s in normal at most 10 s
Payment (banking services and confirming payment.)	a fixed 20 s in normal at most 45 s

Thus, testing and observation is needed. The average response time should less than 8.5 seconds in 95% situations.

Workload

Mainly the background process cause the workload to the website. The workload will greatly affect the response time to the system so it needs to be well defined. The codes behind website's functions should be clean and short enough to enhance the efficiency. The backup procedure may interrupt the response time seriously, thus, it should be processed overnight instead of day time. A workload profile helps the system to define a static workload.

Scenario	Daily Total	Pages	Process Time(each)
Normal Page Browsing	~3000	Search, Post, Course List, Event	1-2 seconds
Login	~1000	Login	5-10 seconds
Payment	~100	Course Payment	20-45 seconds



The peak workload is observed as above. The night time is usually having low workload comparing to day time. In the case of whole month or year, holidays may have more workload since students and adults may access to eBoard finding their favourite courses when they have spare time. The definition of workload helps the system to adjust the jobs for a stable and efficient service.

Scalability

A good scalability definition can lengthen the lifespan of the website. Since unexpected growth of popularity may impact system workload. System may suffer from serious delay or even being shut down. A testing and measurement is needed. Due to the economic concern, simulation model is used to test the scalability.

Years after system opening	Population	Response time (average)
First year	200 -500 max : <= 2000	normal : 8 s upper bound : 15 s
Second - third year	1000 - 2000 max : <= 5000	normal : 8.5 s upper bound : 20 s
Years afterwards	max : <= 10000 varied a lot	normal : 8.5 - 9 s upper bound : 25 s

Platform

Website is usually cross-platform for the operating system. The major concern is whether the website can be display normally in different browsers or not. A simple observation can easily test the results. To prevent it, the .css and style of the website should be common and work fine. Hence, it prevent the website from display issues.

Support browser (first priority):

- Chrome, Safari, Firefox, Internet Explorer etc.