Project :

KMITL-simple-event-bulletins-board

Team : My-name-is-TOM

55011355 Suradit Keereekaew

55010895 Permpon Kurung

Class: Object-oriented Analysis and Design Fall 2014 @ KMITL

Post-second year, Computer Engineering, B.Eng. Semester 1/2557

King Mongkut's Institute of Technology Ladkrabang

Table of Contents

Introduction

Class: Object-oriented Analysis and Design Fall 2014 @ KMITL

Project : KMITL-simple-event-bulletins-board

Team : My-name-is-TOM

55011355 Suradit Keereekaew
55010895 Permpon Kurung

Abstract

Recently, the way to access the internet is so much that make us comfortable more than in the past. This fact makes people more likely prefer to get the news from online media. We see this trend and have idea to build KMITL-simple-event-bulletins-board to replace the board in real world that which is not get enough attention and make a lot of spending. In addition, we design the KMITL-simple-event-bulletins-board for everybody who wants to post some content and upload their poster on website in easy interface.

Introduction & Motivation

The reason that we pick this topic to be a project is we see that the poster and leaflet take too much time and cost. And when the event is expiring these items will not be disposed in right way. So that we want to design some website that can announce and update the KMITL's news. Moreover, we need to get news from anyone who wants to post it because we think that there are some people that want to do it. But they don't have opportunity.

Related works

http://www.reg.kmitl.ac.th: official public relations website for KMITL

http://www.contestwar.com: public relations website for every contest

However, these website need permission to publicize the news. But in our project, we design that everyone can post on our website without permission.

Requirement

Functional requirements

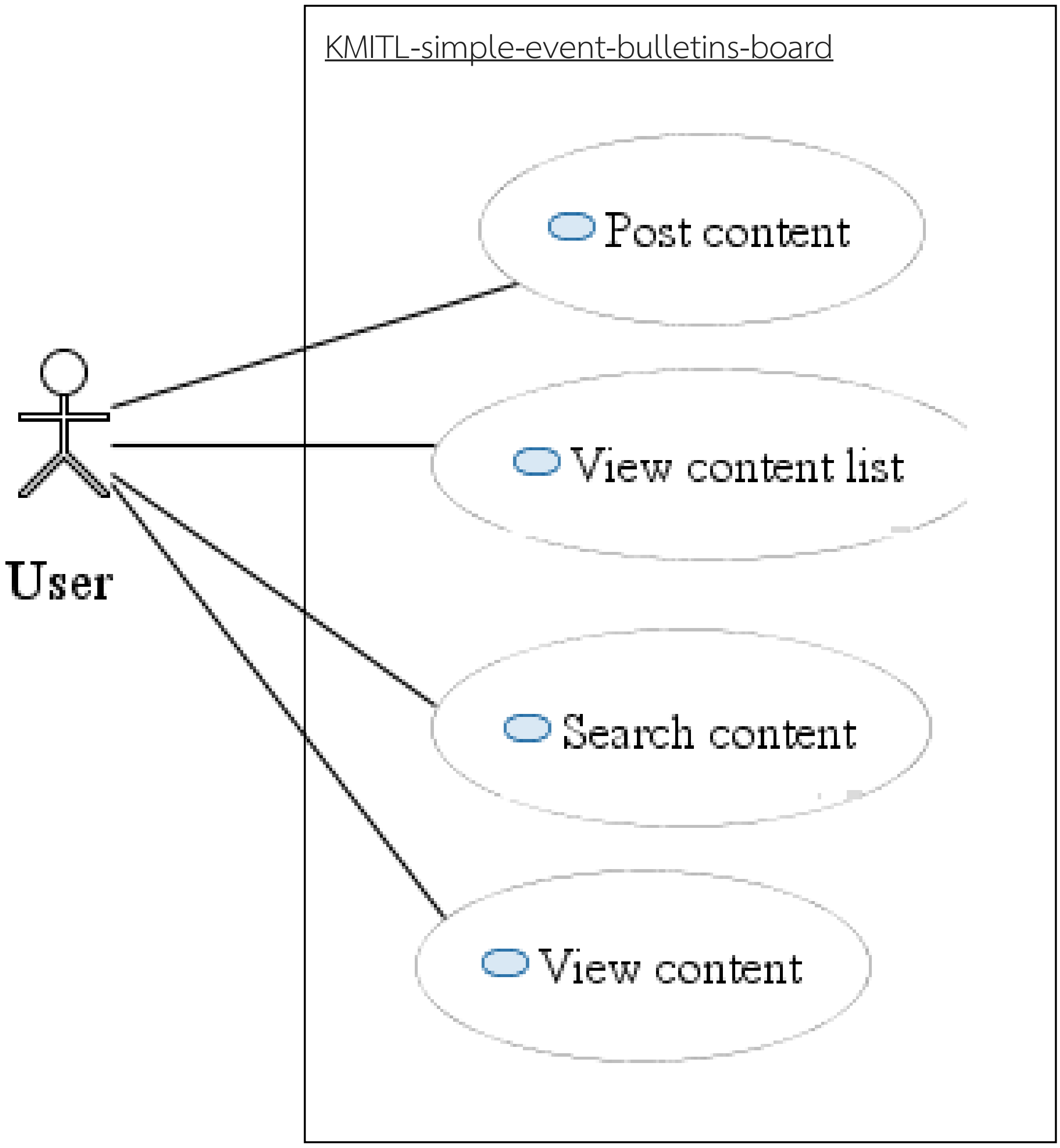
- Everybody can view content by accessing from event's list.
- Automatically hide out of date event in event's list.
- User can search event from keyword.
- User can post KMITL's event.
- User must post their event only by provided form.

Non-functional requirements

- Responsive web page
- Function support on web browser on computer and mobile phone
- Simple pattern interface for post event
- Few content on page to make few confusion
- Few decorated item ,few data be transmitted

Use case

Use case diagram



Use case specifications

Use case name : View content

Use case purpose: Providing the page that actor request with information from database**.**

Assumptions: Actor call this from content list**.**

Primary Scenario :

A. Actor select content from list.

B. System get information from database following selected event.

C. System arrange content in pattern.

D. Actor get information page from system.

Use case name : Post content

Use case propose : This use case will provide actor to publicize their information.

Preconditions: Actor sign in successfully (have cookie save in user's browser)**.**

Postconditions : New event announce on homepage**.**

Primary Scenario :

A. Actor fill title , description due date in form.
B. Actor upload poster picture.
C. Actor submit form and picture to server.
D. Server get data and save in database.
E. Send actor back to homepage.

Alternate Scenario :

Condition triggering an alternate scenario:

Condition 1 : Actor don't have picture to upload.

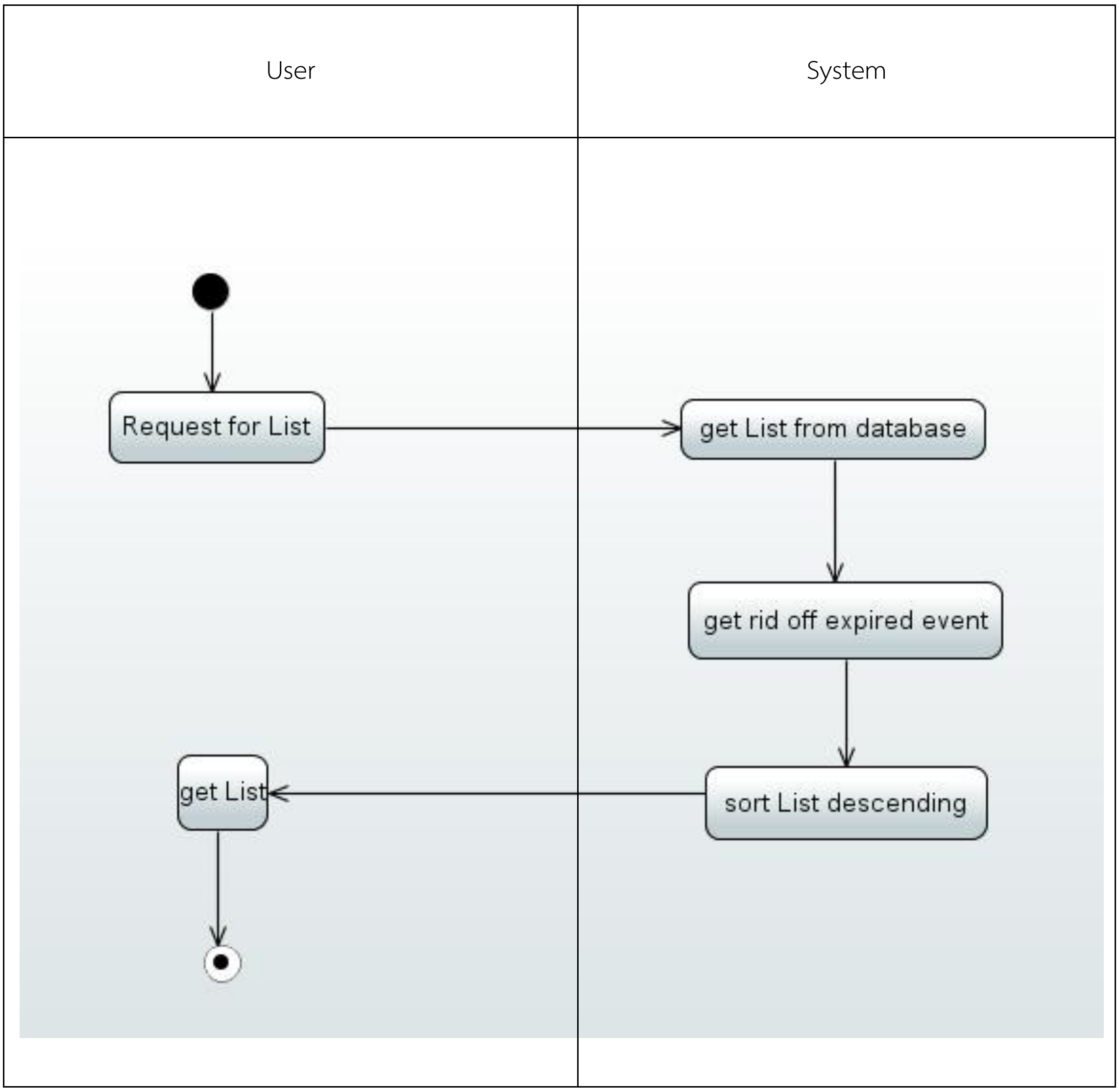
C1 Actor can submit form with out poster

Condition 2 : Don't have title and due date.

D1 Actor get error massage and back to step A again.

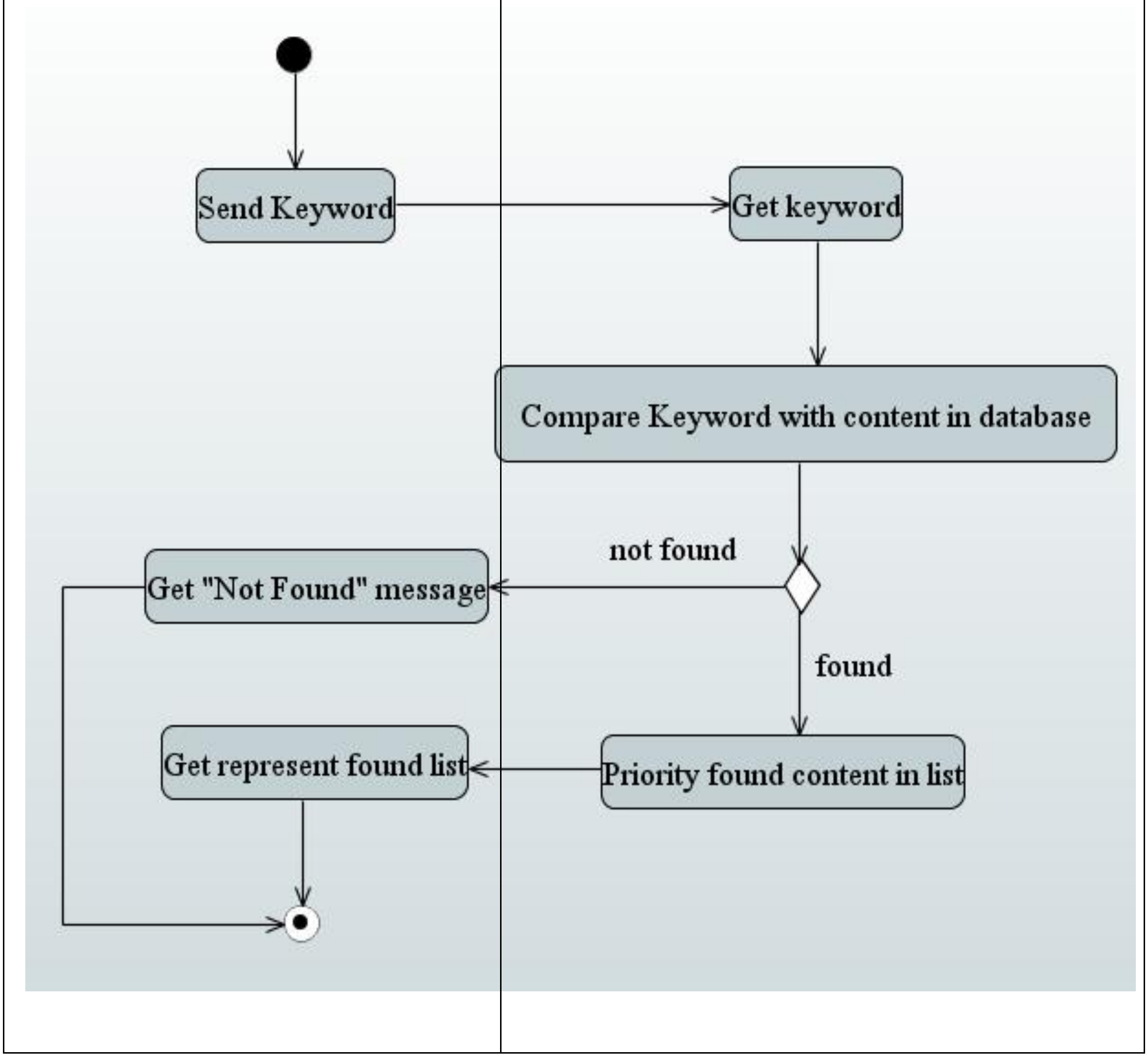
Activity diagrams

View content list 's activity diagrams



Search content ‘s activity diagrams

User	System



Analysis & Design

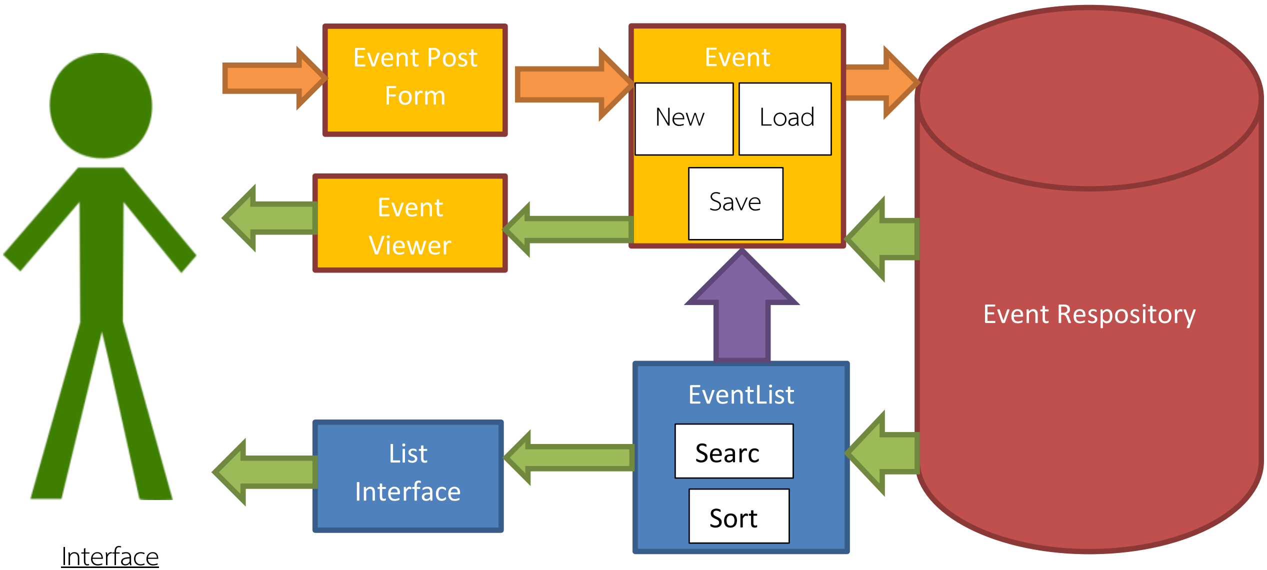
Problem Analysis

We have analysed problem and separate in two part , list and event , because both of them are important logic that our software have.

Component & Abstraction

- Event : contain with title, description, poster image destination, post date, due date and author. This component can be made new one , save or load
- EventList : contain with array of event object that can be search or sort.

Application Architecture



Interface

- Event Post Form : Interface that give users a form to add new their event to database.
- Event Viewer : Interface that provide event's detail requested by users.
- List Interface : Interface that provide event's list to users.

System

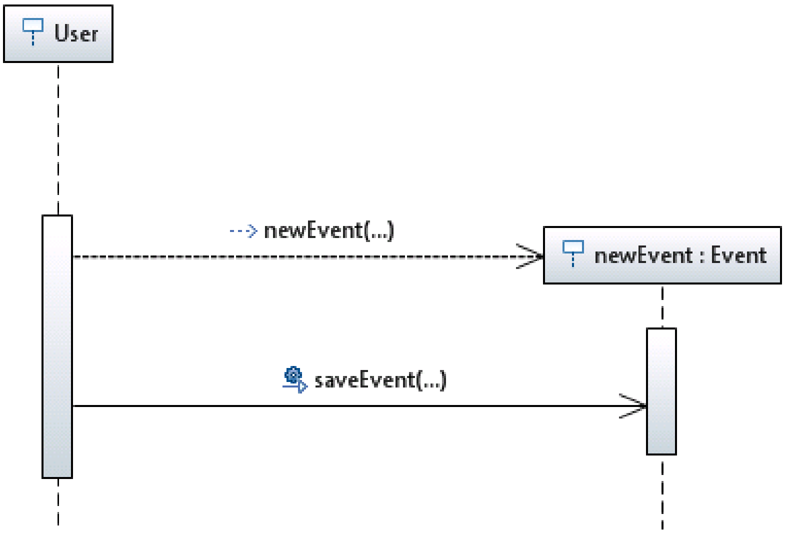
- Event: Component that load and store event's detail .
- List: Component that get data list from Event Respository and search or sort following request.

Subsystems / Components

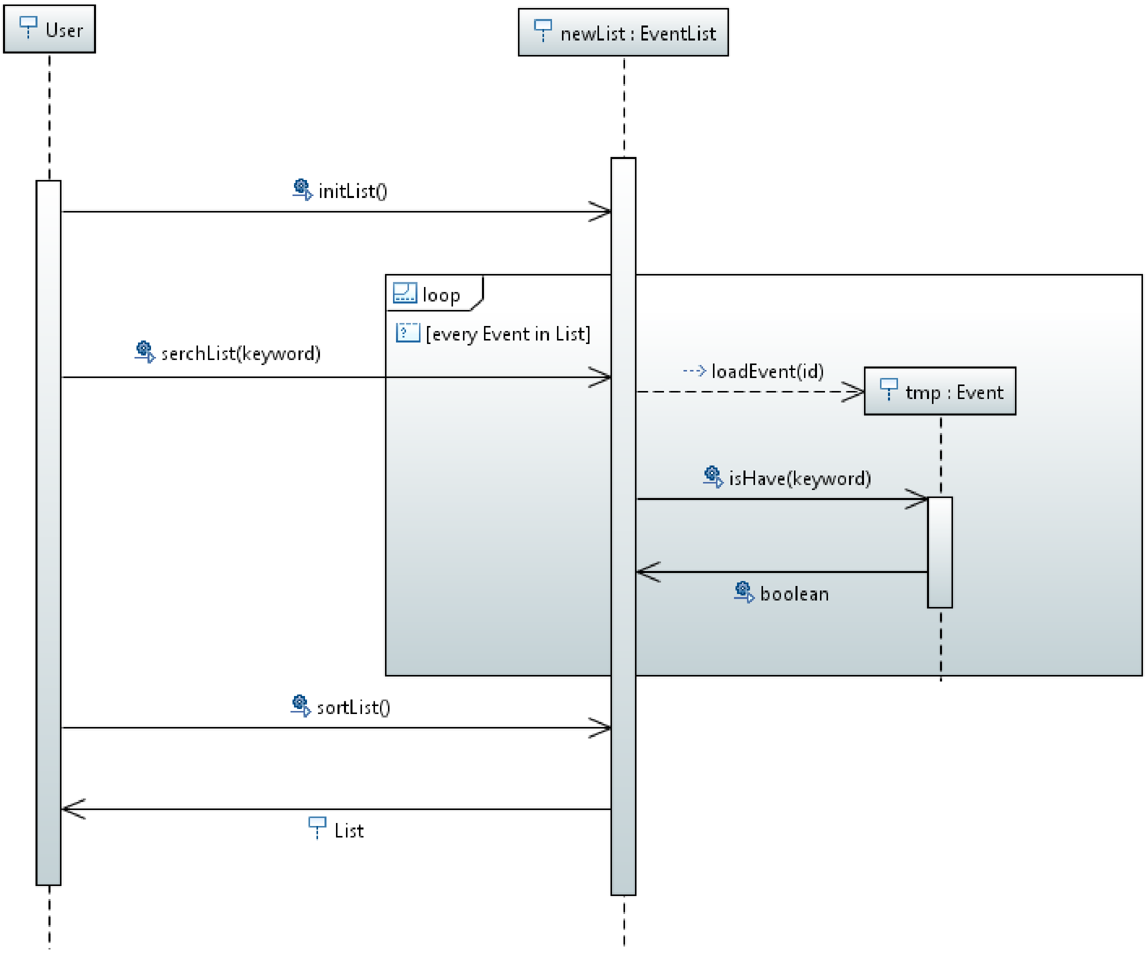
- Event
    - Resposibility :
        - Save Event data from user’s form to database.
        - Get Event data from database following request
        - Edit Event data
    - Interaction :
        - Keep data from Event Post Form to Event Respository
        - Get data from Event Respository to show on Event Viewer
        - Update data that edited by author
    - Subsystems
        - NewEvent
        - LoadEvent
        - SaveEvent
- EventList
    - Resposibility :
        - Give data list to users following users’s request
    - Interaction :
        - Get list from Event Respository
    - Subsystems
        - SearchList
        - SortList

Sequence diagrams

Case 1 : User submit post form to server.



Case 2 : User search title of Event's list that match to keyword and sort by descended author

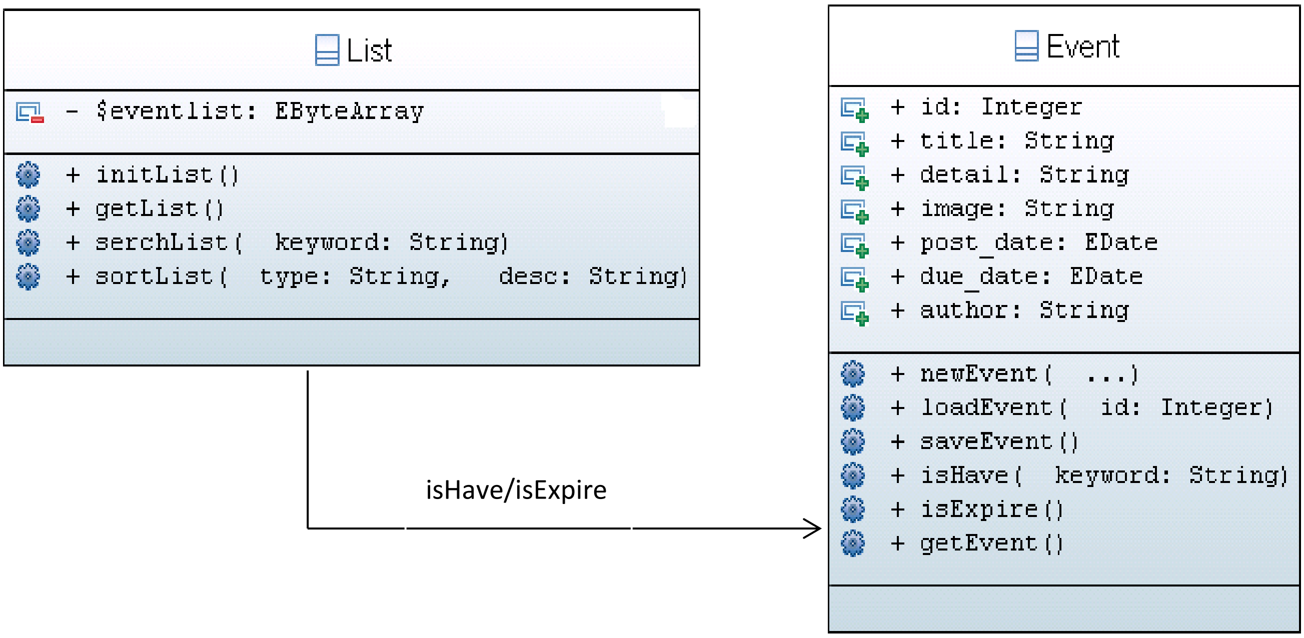


Domain classes

EventList : Class object which represent event data in array

Event : Class object that collect all details of one event.

- isHave(keyword) : return boolean if this event contain the keyword
- isExpire() : return boolean to show if this event is out of date



Deployment

Server : Ubuntu server run on Azure VM

- Web server : Apache + PHP5
- Database : MySQL + phpMyAdmin

Web Application

- Back-end : PHP with CodeIgniter framework
- Front-end : HTML with javascript,CSS,ajax

Implementation Plan

TASK	September	October										November
	27	1	7	12	14	15	16	18	20	23	25	1 6
Implement Event Class		🟧	🟧	🟧								
Implemant EventList Class		🟧	🟧	🟧								
Implement User Class			🟧	🟧	🟧							
Implement Admin Class							🟧	🟧	🟧			
Build Unit Test											🟦	
CONTOLLER												
Connect Event List Interface with EventList Class						🟪	🟪	🟪				
Connect Event Viewer with Event Class						🟪	🟪	🟪				
Connect Post Form with Event Collector							🟪	🟪	🟪			
Build Search and Sort System								🟪	🟪			
Build Sign Up System									🟪	🟪		
Build Sign In&Out System									🟪	🟪		
Connect User Panel with User Class									🟪	🟪		
Connect Moderator Interface with Admin Class											🟪	
VIEW												
Build Event List Interface			🟦	🟦	🟦							
Build Event Viewer			🟦	🟦	🟦							
Build Post Form			🟦	🟦	🟦							
Build User Panel					🟦	🟦	🟦	🟦				
Build Moderator Interface									🟦	🟦		
Other												
Build & Set up VM on Azure	🟧											
Install All Program on VM	🟧											
Build Integration Test											🟦	
Test both unit and Integration												🟦

🟧	55011355 Suradit Keereekaew
🟦	55010895 Permpon Kurung
🟪	Both member

Progress

TASK	Progress	Note
MODEL		
Implement Event Class	Complete	Store in : MVC\models\ event.php
Implemant EventList Class	Complete	Store in : MVC\models\ eventlist.php
Implement User Class	Complete	Store in : ● MVC\models\ authen.php ● MVC\models\ encryption.php
Implement Admin Class	Complete	Store in : MVC\models\ admin.php
Build Unit Test		
CONTOLLER		
Connect Event List Interface with List Service	Complete	Store in : MVC\ controllers\ homepage.php
Connect Event Viewer with Event Class	Complete	Store in : MVC\ controllers\ show.php
Connect Post Form with Event Collector	Complete	Store in : MVC\ controllers\ post.php
Build Search and Sort System	Complete	Store in : MVC\ controllers\ fulllist.php
Build Sign Up System	Complete	Store in : MVC\ controllers\ register.php
Build Sign In&Out System	Complete	Store in : MVC\ controllers\ login.php
Connect User Panel with User Class	Complete	Store in : MVC\ controllers\ homepage.php
Connect Moderator Interface with Admin Class	Complete	Store in : MVC\ controllers\ admin.php
VIEW		
Build Event List Interface	Complete	Store in : ● MVC\views\ eventlist.php ● MVC\views\ fulleventlist.php
Build Event Viewer	Complete	Store in : MVC\views\ eventviewer.php

Build Post Form	Complete	Store in : ● MVC\views\ posting.php ● MVC\views\ uploading.php
Build User Panel	Complete	Store in : ● MVC\views\ loginpanel.php ● MVC\views\ register.php ● MVC\views\ userpanel.php
Build Moderator Interface	Complete	Store in : ● MVC\views\ adminviewer1.php ● MVC\views\ adminviewer2.php
Other		
Build & Set up VM on Azure	Complete	Acess via domain : kmitl-bbs.cloudapp.net
Install All Program on VM	Complete	Acess via domain : kmitl-bbs.cloudapp.net
Build Integration Test		
Test both unit and Integration		

Test

Directory :

Test 1

Test 2

Evaluation

Experiment1

Goal of the experiment:

Measurement metrics:

Setup and methods of your experiment:

Experimental results:

Conclusion:

Experiment2

Goal of the experiment:

Measurement metrics:

Setup and methods of your experiment:

Experimental results:

Conclusion:

Conclusion

According to our progress, we can build KMITL-simple-event-bulletins-board website with requirement such as user can see list without expired event, can search list and can post event with provided form.

In our design, we use concept of object-oriented for abstraction and component. We have two important class in business logic, EventList and Event. Moreover, we have used the MVC pattern for out architecture.

We coded in PHP with SQL database and tested them to make sure that our system is working right. Then we have final progress with end-user trying. In the conclusion we can say that our website can work following our requirement and able to reuse them to improve or add some extra requirement because of object-oriented design.

Problem

- Search algorithm is only find match title.
- Sort algorithm is slow for large data.

Limitations

- Webmaster can only remove data out from database.
- User must have poster that fit to providing pattern.

Development in further

- Use cache webpage for improve accessing speed.
- Extend the way that user can decorate their event.
- Add user data about themselves ,avatar and contact.

Bibliography

HTML/SQL knowledge	: http://www.w3schools.com/
PHP string encoding decoding functions	: http://www.ireallyhost.com/kb/other/194
codeigniter	: http://www.codeigniter.com/