FYP 17008
E-poster board

Interim Report

Luo Kairen
Supervisor: Prof. Lau Francis
The University of Hong Kong
Department of Computer Science

January 21, 2018

Abstract

E-poster board is a web based content management system (CMS) designed for managing a large amount of digital posters with configurable policies. In this report, a new e-poster board system that utilizes powerful programming technique such as MVC, ORM and rest API will be implemented. The goals of this e-poster board system are to provide a flexible and user-friendly way of providing interactive poster content to audience, and to promote information sharing and to enhance social engagement between poster readers and poster providers.

Keyword e-poster touchscreen CMS web

Acknowledgement

I would like to express my special thanks of gratitude to my final year project supervisor Prof. Lau Franics who gave me the invaluable opportunity to do this FYP project on the topic e-poster, which helped me learn software system development and as well as my Technical English class teacher Mr. Patrick Leung who helped me improve my academic English skills.

Contents

List of Figures

List of Tables

Chapter 1

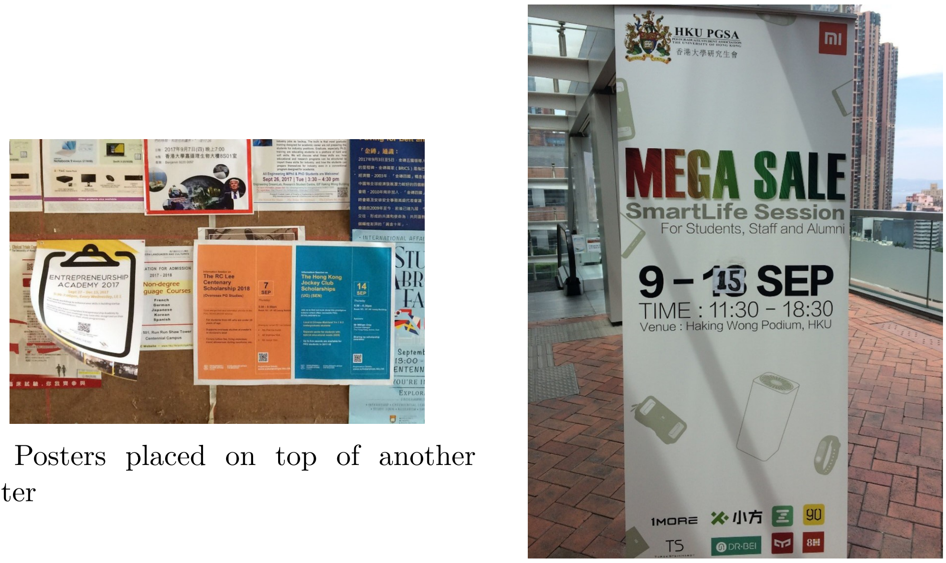
Introduction

1.1 Background

Sharing information using a large display panel within an institution has become a common practice in both private and public space.[1] Delivering event information and notices through a digital media is far better than paper posters in terms of management and speed. Also, public digital displays save the effort for poster readers to actively search for information from social media or emails and therefore they help readers focus on closely related information.

However, most existing displays are designed for distributing content in a broadcast manner and few of these displays have interactive capacity such as sorting contents according to their categories and discovering further information based on keywords. This one-way sharing channel hinders the distribution of interactive multi-media content and stop readers from actively exploring events that meet their personal preference. Due to the constraints of display size and lack of interactive features, readers may find that information on the non-interactive e-poster board too generic and irrelevant. The lack of control over the type of information and its breadth and depth make reading traditional poster become an exhausting and inefficient job, and this prior investment of time and energy prevents reader from getting involved into social events they may be interested in.

In the last few years, thanks to rapid growth in smartphone market, demand for touch screen panel with high resolution and high precision has greatly increased. Price of touch screen has been driven down by this trend in the meantime.[2] It is now practical to use large affordable display panel with touch capacity to replace traditional poster board and display with no touch capacity to provide better information service for organization that


ENTREPRENEURSHIP ACADEMY 2017
The RC Lee Centenary Scholarship 2018
The Hong Kong Jockey Club Scholarships
HKU PGSA
MEGA SALE
SmartLife Session
For Students, Staff and Alumni
9 – 15 SEP
TIME : 11:30 – 18:30
Venue : Haking Wong Podium, HKU

(a) Posters placed on top of another poster

(b) Method used to update a physical poster in campus

Figure 1.1: Pictures to illustrate problems on physical posters

need an attractive way to promote events or activities.

1.2 Motivations

First, physical posters are hard to be organized and managed. Unlike digital posters which space and printing cost are not a major concern, physical posters have the disadvantage of inefficiency in term of poster board space usage and difficult arrangement for posters with various size and different topic. As shown in figure 1.1a, it is quite common to see when the number of posters become large, it is unavoidable to place posters over one another. Without a automatic solution to organize posters according to their content and submission time. It makes searching for posters a time consuming task for viewer to find latest and relevant information.
Second, information on physical poster tend to be inconsistent with the official source. Once a posters are printed, updating them is labor intensive and it usually cannot be done in a short amount of time. The lack of resource to do full scale update as well as the time gap between updating cause out-dated information on physical posters. An example of mega sale poster is shown in figure 1.1b. The inconsistent problem can be solved if posters are stored in digital format and synchronized with a center database.

1.3 Objectives

The aim of this project is to build an electronic poster board that can promote information sharing and enhance poster viewing experience. It is hoped that through the use of the new e-poster board, the social tie between poster readers and poster content providers will be greatly strengthened and readers could feel that it is more intuitive to be informed of new events and get involved in the community.

1.4 Scope

This project serves as a content management system(CMS) behind any arbitrary e-poster board hardware with display screen and touch capacity. This system will provide a workflow for content provider to create and submit interactive poster,an web based poster viewing interface that follow rules and policies designed by system administrators and a data analysis platform for finding out reader viewing pattern.

1.5 Outline of the Report

The remainder of this report proceeds as follows. First, some leading digital poster solutions will be analyzed to explain why they are not suitable for our usage scenario. Then an implementation of e-poster system that avoid drawbacks in previous works together with numerous additional enhancements will be described. After that, it would demonstrate some difficulties and limitations encountered during the system development phase and their mitigation solutions.

Chapter 2

Related Works

There are several existing e-poster systems on the market. A brief analysis will be conducted on squareVIEW, iPosterSessions and ePosterBoards.

2.1 squareVIEW Digital Posters by Digital Media Systems

SquareVIEW is digital poster hosting solution provided by Digital Media Systems for different scenario such as ”displaying products & adverts, menus, employee communications and welcome messages”.
The reason I pick squareVIEW as the first example is that there are many digital posters solutions that are based on ”Android Freestanding Digital Posters” embedded system which is a 30-50 inch standing touchscreen monitor that has been widely used in Hong Kong. In order to avoid duplication, I selected that most representative provider to discuss.
See Figure 2.1. They are android freestanding digital posters for displaying poster in picture, video and powerpoint format. When poster providers need to update poster content, they can transfer files from a USB stick to the digital poster device, therefore squareVIEW Digital Posters are usually used for files that does not need to be updated over the internet or internet connection is not accessible. [3]

2.2 iPosterSessions by aMuze!

The iPosterSessions is commercial Software as a Service e-poster system provided by aMuze!. From the description of aMuze!, iPosterSessions is built for scientists and scholars to present their research posters electronically and

Figure 2.1: squareVIEW Digital Posters

interactively.[4] It allow users to host poster in form of webpage. The system support multimedia playback such as image, video and audio. See Figure 2.2. As it is intentionally designed for aMuze!. Each e-poster screen will only display one poster and there is no navigation functionality to switch to another poster. Besides that, the layout of poster that can be used on iPosterSessions is limited to a set of predefined template. Users need to build their poster with iPosterSessions' template and users cannot reuse posters that are designed using other software since they are incompatible with the system.

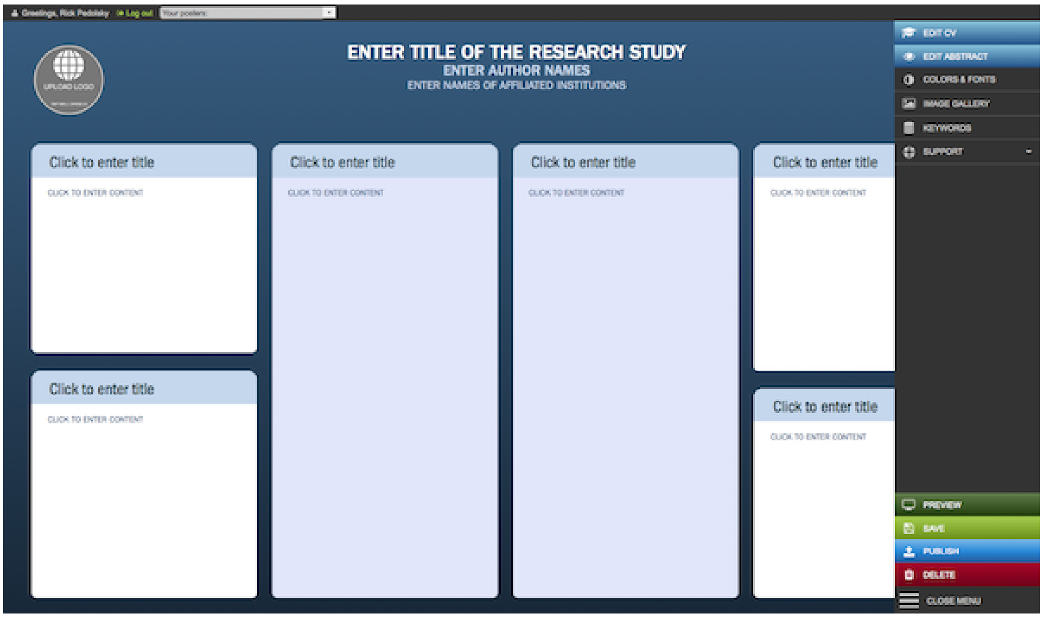


Figure 2.2: iPosterSessions

2.3 ePosterBoards by ePosterBoards LLC

ePosterBoards LLC is a event technology service company based in greater Boston area. Their e-poster solutions ePosterBoards is a all-in-one system for rental which hardware and software work as a whole. Their solution features responsive design available on all devices, analytics for tracking ePosters viewed and user-friendly interface.[5] See Figure 2.3. Each e-poster screen can only show one poster at a time. For e-poster format, ePosterBoards only accepts pdf files that are created based on predefined template, and their system does not support video or interactive elements.

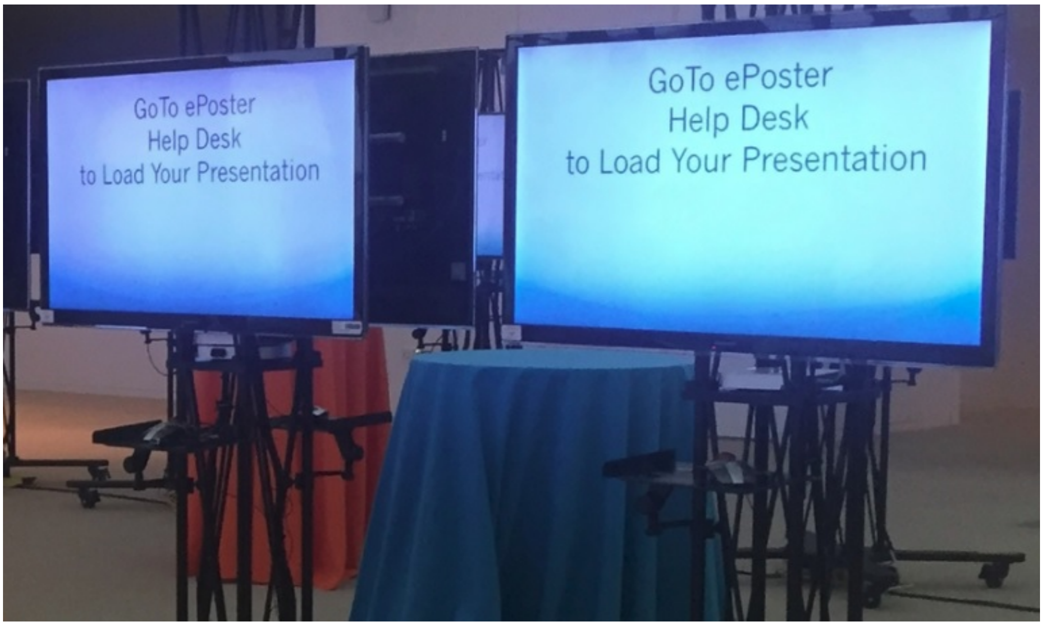


Figure 2.3: ePosterBoards

2.4 Limitations of existing products

There are several drawbacks for e-poster products on the market based on observation from section 2.1,2.2 and 2.3. Details are list in Table 2.1

First, most commercial solutions don't provide self-host functionality. Considering organization may not want to share their internal event schedule with an external party. Solutions without self-host functionality may not be suitable for internal use. Second, some e-poster system such as squareVIEW

Functionality	squareVIEW	iPosterSessions	ePosterBoards
Self-host	✓	X	X
Interactive Support	X	✓	✓
Management over Internet	X	✓	✓
Multiple posters on one device	✓	X	X
Support for standard format eg, Portable Document Format(pdf)	✓	X	✓

Table 2.1: Comparison of different e-poster solution on the market

Digital Posters relies on hardware that does not have touch screen panel and internet connectivity, such that their poster contents would not be able to interact with user. Since they are not internet connected, this makes content management very difficult as well. Lastly, some existing e-poster systems such as ePosterBoards address conference needs only which means each e-Poster is displayed on one terminal as space is not the primary concern, therefore support for multiple posters on single screen is missing or limited. Due to the constraints of space and building infrastructure , not being able to display multiple posters one screen makes e-poster adaption very hard for low-budget users.

In short, after research, it is found that no existing product can fulfill all requirement

Chapter 3

Methodology

This chapter serves as an overview of different system designs used in the system. Firstly, system's architecture that includes an abstraction of software implementation and communicate routes between system parts. Then, the system is separated into components according to their functionality. This both helps system implementers understand the role of each component and it also helps system implementers communicate using same set of naming convention.

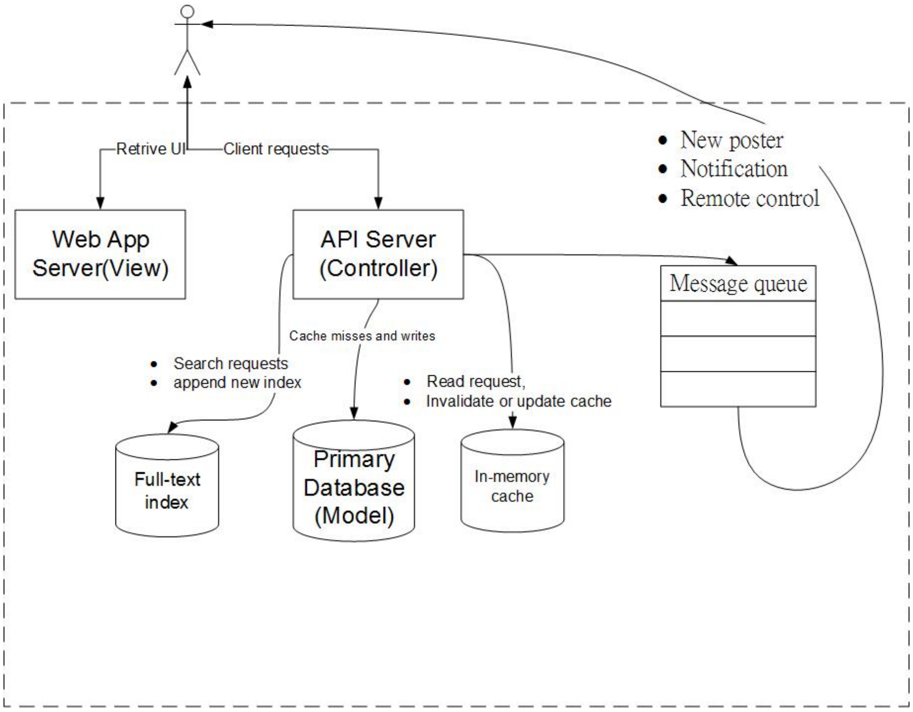


Figure 3.1: Architecture of different infrastructure parts

3.1 System Architecture

There are several considerations before starting building the system. Since the purpose of this project is to demonstrate potential of a modern e-poster system, design of the system focuses on usability, maintainability and extensibility. In the first iteration, maximum usability is the foremost thing to consider. A system working prototype with monolithic structure will be built, and based on users' feedback, the system could quickly evolve to address possible issues and additional requirements. In the next stage, as shown in Figure 3.1, front-end user interface and back-end system will be separated into two parts and communication between them are replaced by API interface instead of direct function call. This helps separate concerns between drawing a user-friendly user interface and building a robust back-end system. Decoupling of unrelated parts help make the system to be more maintainable and extensible.

3.2 System Components

All major parts of the system are listed in Figure 3.2. The general system component design would serve as a guideline for separating the system into different tier which can be implemented and tested independently. They are namely Presentation Tier, Application Tier, Data Analysis Tier, Storage Tier, Admin Console. For Admin Console, this system component would potentially interact with all four tiers, so it is shown on the side of the diagram.

For Presentation Tier, it is responsible for rendering interface display and translate interface interaction into system requests that can be processed by Application Tier. For Application Tier, its job is to do the heavy lifting for cooperating resource from system infrastructure and third party service to provide actual service. Data Analysis Tier acts as a middleware. Its duty is simple, which is to analyze requests from Application Tier to Storage Tier, and generate feedback data back to Application Tier. Storage Tier is an encapsulation of ORM and other utilities that will interact with the database infrastructure.

3.3 Development Environment Setup

The system will be developed and tested under following environment and unless otherwise specified, all system performance evaluation and optimization will be based on this setup.

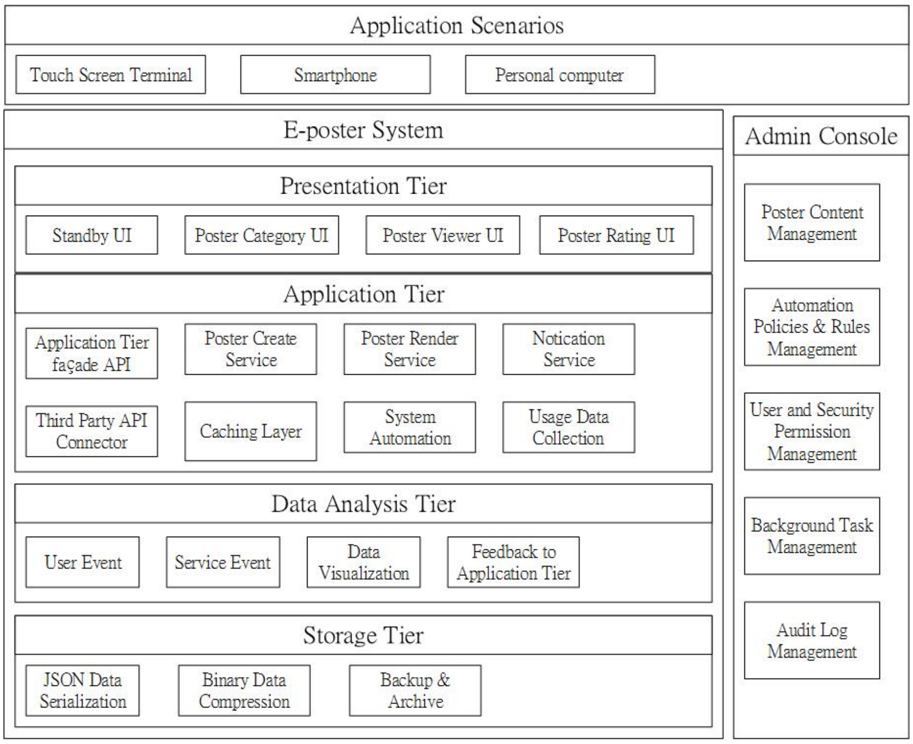


Figure 3.2: An overview diagram to illustrate system components and their hierarchy.

- Hardware: virtual machine(VM) that has one core of Intel i7 CPU and runs on Openstack Icehouse distribution(HKU FYP server)
- Operation System: Ubuntu 14.04 LTS Server
- Backend Platform: Spring Boot 2.0.0 M7
- Frontend Platform: React v16.0.0
- In-memeory cache: Redis 4.0 stable

3.4 Designing a new e-poster format

To incorporate rich multimedia and maximum interaction features into the e-poster system, a new e-poster format that has good compatibility with well-known multimedia files and is easy to create is essential for the success of the system. This poster format should be easily rendered and converted into browser compatible form. Also, it is better to be stored in plain text form with meaningful scheme to enable searching and versioning.

Chapter 4

Implementation details

This section serves as an example of a purposed implementation for the e-poster system design. This e-poster board system comprises a user interface that suits terminals with different screen size and a backend server running in service provider's machine, which provides posters content data. The backend server can also response to terminals' requests for different type of services.

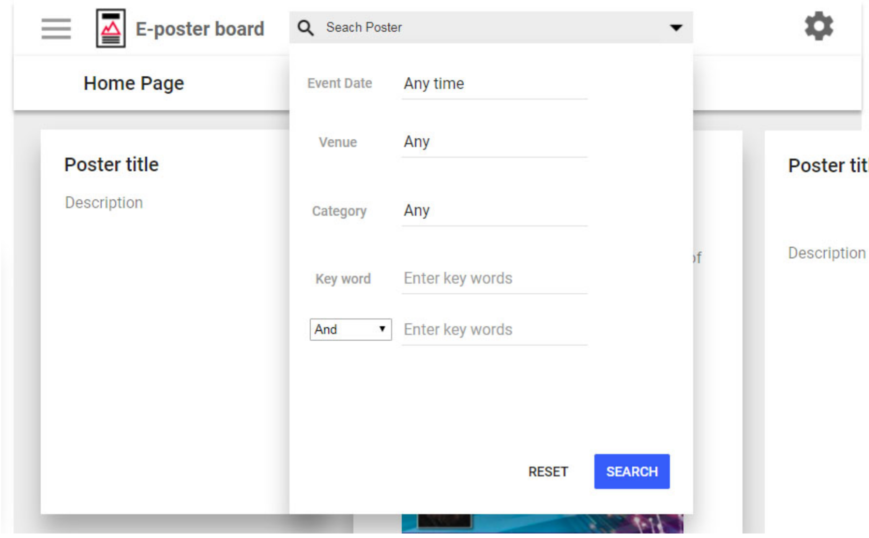
Beside the user interface and backend server which are two main parts of system, several other system communication standard and system parts will be discussed separately. They are the purposed poster format, real time board update protocol, and the recommender system for providing poster suggestion feature.

4.1 Terminal User interface

The terminal user interface is a web based single-page-application together with multiple predictable state containers and the interface's functionalities include retrieving current posters information on the server and starting WebSocket workers to subscribe any subsequent poster modifications or broadcast events.

Information exchange between the terminal and the server or internal communication occurs by looking up corresponding API interface and initializing HTTP request in predefined message format. A consistent way to exchange data is convenient for analyzing network communication and making changes to the state container which reduces the chance of getting into unforeseen system status, which enhances user interface stability.

To optimize user experience on a touch screen, a user interface design principle called material design is putted into practice. According to the description on Material Design's homepage, Material Design provides a framework that provide a unified experience for cross-platforms and devices with different screen sizes. [6] Our system adapted the design principle as well as many standard components from Material Design as you can see in Figure 4.1, 4.2, and 4.3 which shows the system user interface on poster searching, poster viewing and poster recommendation respectively. The aim of adapting material design is that the familiarity of material design may help the first time user navigate the system and it also make the system responsive to different touchscreen size.



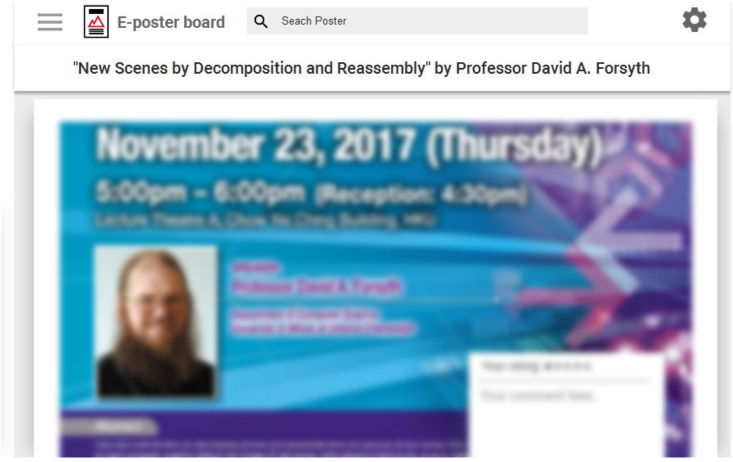
Event Date: Any time
Venue: Any
Category: Any
Key word: Enter key words
And: Enter key words

Figure 4.1: User interface for searching posters

4.2 System backend server

4.2.1 Database design and set-up

The system's database is managed by a object-relational mapping tool Hibernate which supports all major database vendors and acts differently to expose a high level programming interface. This saves an institution from worrying about the compatibility issue with low-level database management. The overview of database structure are shown in Figure 4.4. This ER diagram explains the relationship between an e-poster with other database entities. Each entity is stored in a separated table which is serialized and stored in primary database and each record in a table has a unique key which


E-poster board
Seach Poster
November 23, 2017 (Thursday)
5:00pm - 6:00pm (Reception: 4:30pm)

"New Scenes by Decomposition and Reassembly" by Professor David A. Forsyth

Figure 4.2: User interface for viewing posters

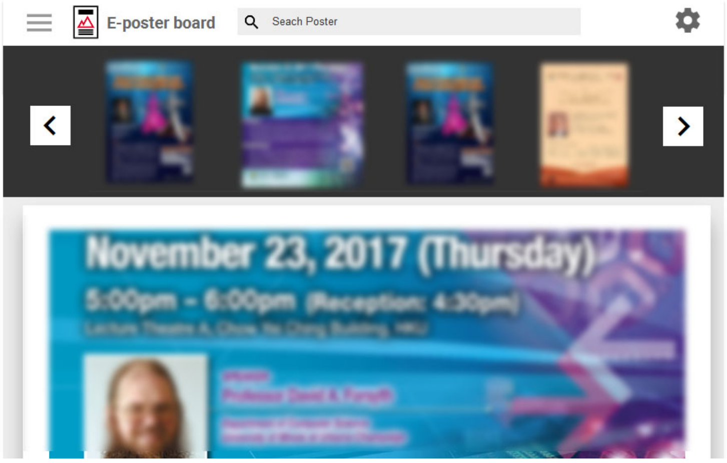

E-poster board
Seach Poster
November 23, 2017 (Thursday)
5:00pm - 6:00pm (Reception: 4:30pm)

Figure 4.3: Standby page for providing poster recommendation

is denoted as PK on the diagram.

Poster table is used to store e-poster items. Poster entities store poster content of desired poster format in poster_content attribute together with a rendering snapshot to be shown on the poster showcase layout. Each poster has zero or more event entities which would be essential for providing map navigation and event registration service. Event's venue information is also very important for providing poster recommendation to specific target group. Details about the recommending system will be discussed on section ??.

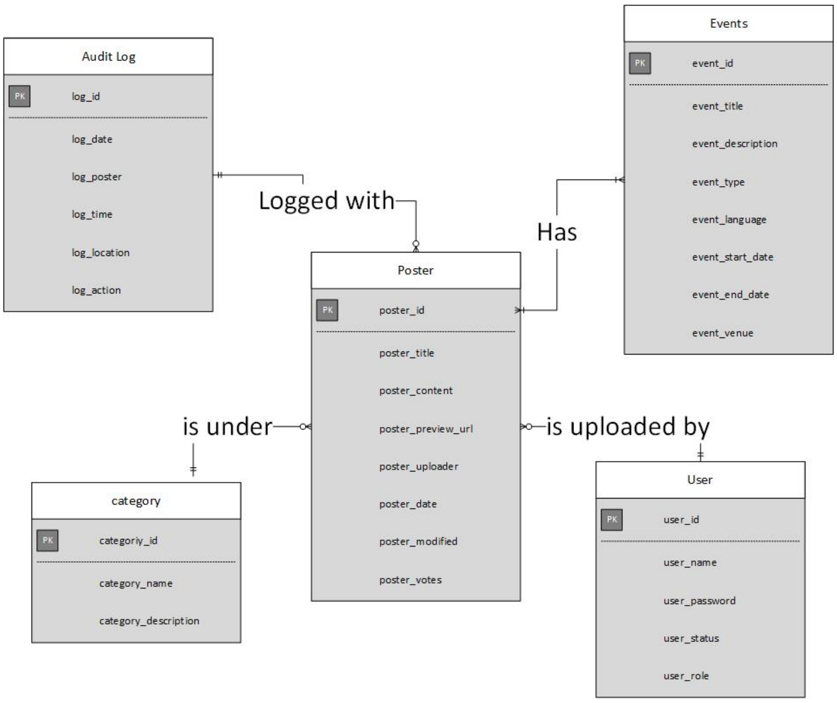


Figure 4.4: An ER diagram of the database setup.

4.3 Purposed Poster Format

There are many potential candidates of poster format, namely plain pdf format, plain html format, pdf format with HTML element layer, pdf converted into image with HTML element layer and pdf converted into Scalable Vector Graphics(SVG) with HTML element layer. Each candidate has been used in industry for many years, especially plain pdf and plain html are simple to implement and have best compatibility with embedded digital devices. Previous works squareVIEW Digital Posters and iPosterSessions are both built based on these two poster format.

However among all these formats, the last option pdf(svg) with html elements is chosen because pdf(svg) doesn't have major defects like other formats in common e-poster usage scenario. See Table 4.1. For plain pdf format, it has a major disadvantage which is poor interaction with the web platform like not being able to play audio and video since Adobe pdf plugin is built on a browser plugin architecture NPAPI which has been phased out due to security reasons.[7] [8] [9] For plain HTML poster format, this format is best suited for expressiveness and performance in web platform. The only drawback is that creating a HTML file that work correctly and look like a pdf poster file a never a easy task. A slight better version of plain pdf would be augmenting pdf with html elements. This brings the power of interaction into pdf file, but it also introduces the cost of incompatibility with mixing pdf and HTML contents.

So in order to combine that easiness of typesetting using pdf and the interaction capability of web elements, a compromised solution would be converting pdf into a web elements that can still perverse pdf's content. A primitive implementation would be converting the pdf file into a binary image format which would is one of the most supported element of the web standard. However using a image format to store the poster content is undesirable as it is slow to transfer over the Internets and it is costly for storage.

Among all other image file formats, vector graphics format svg stores image information with polygons and node attributes which is an acceptable replacement of binary image file in presenting a digtal poster where majority of the poster content are about text and geometry graphics. Also, svg format can be converted into HTML canvas element which make svg format fairly easy to be adapted on web platform.

The working principle behind this svg-html hybrid approach is fairly simple. First traditional poster file like an pdf file that may contains original contents like text and image is analyzed. Meta data like position, size, color and font used by a piece of text can recovered by referencing Portable Document Format (PDF) 1.7 specification provided by Adobe. Based on these meta data and multimedia resource extracted from the source file, they can be restructured and rebuilt with svg file format. Then additional html elements that provided interactive functions like event reminder and map navigation are inserted on the top of svg file, and css stylesheet is used to make sure these html elements appear in correct position as shown in Figure 4.5.

Table 4.1: Comparison of different e-poster format choices

	Plain pdf	Plain html	pdf with html element.	pdf(image) with html element.	pdf(svg) with html element.
searching	normal, pdf's text content can be extracted and indexed, but the quality may not be satisfactory.	best	normal, same as plain pdf	normal, same as plain pdf	normal, same as plain pdf
interaction	worst	best	good	good	good
performance	bad	best	worst,	bad - good, image's loading performance largely depend on network speed	good
storage	good	best	good	worst	good
compatibility with embedded platform	best	best	worst, behavior of drawing additional HTML element on top of a browser's built-in pdf viewer is unpredictable.	good	good
easy to create and modify	good, may need help from external pdf editor	worst, require lots of technical skill in creating html page.	normal	good, same as plain pdf	good, same as plain pdf

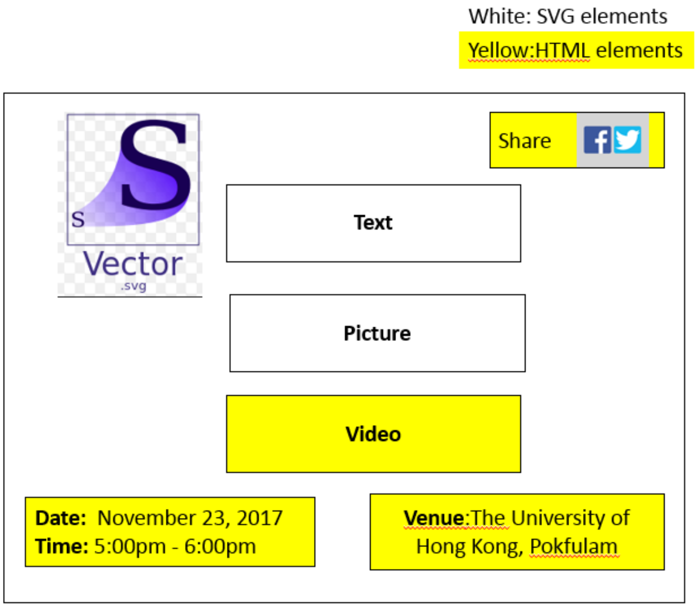


Figure 4.5: Placing HTML elements on SVG file

4.4 Real Time Board Update

As stated in the motivation, one of the key benefit of digital poster is the consistency of information between poster provider and viewer in real time. But unfortunately, the entire web platform is based on a protocol called HTTP(Hypertext Transfer Protocol) which does not allow server to actively send message to its client without client making a request first.

There are several common used methods to achieve data synchronization under this constrain. They are short polling, long polling and websocket. Websocket is chosen in this case as it is much more resource-efficient compared to other ways[10]. A metaphor on the working principle behind websocket is shown in Figure 4.6. When the client successfully retrieved and executed the page that need to be updated in real time, a instruction will be sent to browser's websocket opens a persistent tcp/ip connection with the server, which is like subscribing to a email list. After that, both server and client can communicate via this full-duplex asynchronous communication channel without blocking the main thread. On the other hand, short polling and long polling are like picking up a phone with a third party which short and long refer to the length of duration between each polling request. To approximate the effect of real time, the client side need to keep talking to the other end in order to confirm availability as well as to check for the presence of new information.

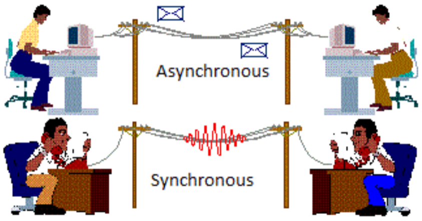


Figure 4.6: Asynchronous vs Synchronous Model

4.4.1 Real time communication API design

To reach real time data synchronization that is capable of performing board state delta update, a message format with special payload that can be understood by both server and client is designed.
There are three type of payload that can be sent by the server, they are

- hello. This message is used to agree on API scheme.
- ping A message from server to check availability.
- pong A message from client to confirm availability.
- error. Return error message on illegal or unexpected operations.
- action. Standard message type for all other operations.

The specification of the Real time communication API is listed as below.
Message format:

```
{
        ”type”:  ”MESSAGE_TYPE”,
        ”ATTRIBUTE1”:”VALUE1”,
        ”ATTRIBUTE2”:”VALUE2”,
        ...
}
```

Possible values for attribute event:

Event	Description
Data sync related	

poster_added	A new poster was added,
poster_change	A poster’s content had been changed
poster_down	A poster was taken down
System administration related	
open_poster	Force switch current screen to participle poster
reconnect	Try to reconnect websocket with a new server address
token_revoked	The JWT token is forcibly revoked to logout current user
Debug related	
show_message	Display a message on the screen
show_terminal_status	Display terminal status on the screen
refresh	Refresh the terminal

4.4.2 Board Data Synchronization Steps

In the first step, client send a HTTP request including Upgrade header to indicate this is a websocket connection, this is known as WebSocket handshake.Once the server that supports websocket protocol receives the request, it send back a acknowledge response with HTTP status 101.
Now a asynchronous communication channel has been established, server send a hello message to the client.

```
{
        ”type”: ”hello”,
        ”verion”:”1.0”,
}
```

If client cannot receive the hello message, a retry process will be started. From the hello message, client know the communication channel is working. Also, it confirms that the API version has been implemented in the client script. If not, the client page will refresh to retrieve the latest implementation from web server.

Step two. server initialize a ping-pong checking procedure. For each minute, server send out a ping signal to all connected clients, and clients reply by pong to confirm it.

```
{
        ”type”: ”ping”
}
```

,and

```
{
        ”type”: ”pong”
}
```

Step three. Server logs down the current status of latest data set. If there is any modification on the data set, a message with payload type action will be sent to the client together with the corresponding action event.

```
{
        ”type”: ”action”,
        ”timestamp”:1516459270,
        ”event”: ”poster_added”,
        ”payload”:{
                ”id”:”1”
        }
}
```

For exception cases. There are many possible reason that real time update will fail. Some of the possible reason include timeout due to heavy workload, design error in implementation. Client can extract those error message to inform its user the exact problem.

```
{
        ”type”: ”error”,
        ”timestamp”:1516459270,
        ”error”:{
                ”code”:0,
                ”message”: ”Timeout”,
        }
}
```

4.5 Recommender System for Poster Suggestion

The recommender system for providing poster recommendation is based on algorithms that meet following criteria. First, it has to balance between performance and recommendation quality. Since the number of posters will keep growing, the system must be scalable for future usage case. Second, prediction results must be reproducible to enable comparison with user's real choices using root-mean-square error(RMSE). This is essential since it provides an optimization direction for improving recommendation quality both automatically or manually. Finally, a potential parameter in the model may include reader's location, current time, posters' vote count, posters' submission time, event venue and event schedule, and the recommendation model must be able to increase the number of parameters and place weighting on its parameter, because some information is more important, and it is hard for a learning algorithm to get the sense of importance without human's help.

Chapter 5

Difficulties and Limitations

5.1 Performance issue on complex poster items

Parsing a huge svg vector file in plain text form can be time consuming, The reason why svg is bloat and of large size usually would be technical decision in pdf to svg converting step doesn't meet our needs. The top priority of pdf to svg converter are to parse all detail of pdf contents and translate them in svg standard. In this process, many unnecessary information like unnecessary attributes, invisible elements and machine generated comments make the svg file very bloat and therefore it is slow to read and process. Since the svg won't be modified or analyze extensively by the system, there is no need to preserve these hidden information. It is safe to remove these unnecessary contents in the converting step.

Also, another preventive method would be setting restrictive policies for file of maximum number of elements or largest file size allowed to be uploaded before processing to the parsing step.

5.2 Performance issue on recommendation system

Due to the information-intensive nature of e-poster file, ranking operation is inflexible to perform using the request-and-response method in real time. Hence, a engineer decision is made to reduce system load by utilizing in-memory caching . After the initial calculating, recommendation results could be stored in in-memory key-value store and later these information can be retrieved from the store to accelerate the whole process. The calculation will only run once an hour to keep ranking information updated, and this simple engineering tactics avoid repeating expensive calculation steps and reduce response time on the terminal end. Although ranking result may not be up to date in real time, it is not a major issue as poster submission and modification are expected to happen in a relatively steady and moderate rate. A good ranking algorithm would not have huge variation in output result when input sources have little change.

Chapter 6

Project Schedule and Deliverables

Date	Task	Status
1st Oct 2017	Deliverables of Phase 1: Detailed project plan Project website	completed
Oct 2017	Find a migration solution for existing posters. Design a poster submission workflow. Design an e-poster board user interface.	completed
Nov 2017	Build poster viewing UI.	completed
Dec 2017	Add more interaction elements to poster viewing UI. Build data analysis platform.	completed
Jan 2018	Build poster application backend. Build full-text search engine.	Working on it Working on it
20st Jan 2018	Deliverables of Phase 2: Detailed interim report	completed
Feb 2018	Build a recommender system	Not yet started
Mar 2018	Testing and debugging.	Not yet started
15 Apr 2018	Deliverables of Phase 3: Final report Finalized implementation of E-poster board	Not yet started

Table 6.1: Project Schedule

To summarize the project progress, the system 's user interface for terminal device and admin console has been implemented as well as an essential set of backend application functions. The full implementation for application logic is expected to be finished within January 2018.

Also the storage tier such as database scheme has been designed and deployed. With a working database, the user interface is now collecting user interaction and user feedback.

In February 2018, we will start working on a recommender system that is capable of suggesting posters to user based on different context such as time and place. The recommendation algorithm will be evaluated with existing datasets on the internet to compare its performance with other recommendation algorithms to see how the extra context information could help user better find information.

Chapter 7

Conclusion

To help community member find out events and notices they interested in, this paper describe a possible implementation of e-poster board system. The system uses a multi-tier architecture and responsibilities of each tier are separated to reduce programming complexity. This report offers contributions in three areas. First, a suitable software architecture for e-poster system is designed, and it is practical for this architecture to handle specific needs in providing interactive functionalities. Second, it introduces a new poster interaction format to enhance user experience. Third, this report provides solutions for technical problems such as poor scalability of complex poster file.

There are several directions for future development and research on improving e-poster board system. One direction would be reducing set-up environment constraint without sacrificing usability. Currently, this poster system require a terminal device with internet and interactive display support. This may seem acceptable in in-door scenario, but it may not be flexible to meet these setup requirement in out-door space where wireless connection and display protection may not be present. A better hardware equipment setup with better out-door space support should be found when internet-of-thing(IoT) devices become more prevalent. Another direction will be developing better tool-set for creating svg file with HTML elements. As svg file has good compatibility with PDF format and the web platform, it is not hard to predict it will be widely used for displaying complicated document file with interaction functionality. A mature workflow both to create and publish web-argumented svg file will be in great need in the near future.

Bibliography

[1] E. F. Churchill, L. Nelson, and L. Denoue, "Multimedia fliers: Information sharing with digital community bulletin boards," in *Communities and technologies*, pp. 97–117, Springer, 2003.

[2] J. Colegrove, "The state of the touch-screen market in 2010," *Information Display*, vol. 26, no. 3, pp. 22–24, 2010.

[3] "Digital posters — electronic signs." `http://www.digitalmediasystems.co.uk/digital-posters/`. Accessed: 2017-09-30.

[4] "Interactive electronic posters for multimedia presentations." `https://ipostersessions.com/`. Accessed: 2017-09-30.

[5] "eposterboards." `http://www.eposterboards.com/conference-setups/`. Accessed: 2017-09-30.

[6] "material design, introduction to principles." `https://material.io/guidelines/material-design/introduction.html#introduction-principles`. Accessed: 2018-01-12.

[7] "The final countdown for npapi." `https://blog.chromium.org/2014/11/the-final-countdown-for-npapi.html`. Accessed: 2017-10-22.

[8] "Netscape-style plug-ins do not work after upgrading internet explorer." `https://support.microsoft.com/en-us/help/303401/netscape-style-plug-ins-do-not-work-after-upgrading-internet-explorer`. Accessed: 2017-10-22.

[9] "Version 52.0, first offered to release channel users on march 7, 2017." `https://www.mozilla.org/en-US/firefox/52.0/releasenotes/#changed`. Accessed: 2017-10-22.

[10] V. Pimentel and B. G. Nickerson, “Communicating and displaying real-time data with websocket,” *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, 2012.