

Software Engineering Group Project

Co-Papers – A Platform to Organize and Share Your Academic Notes

Group Number: 4

Supervisor: Dr. K. Papangelis

Xi'an Jiaotong-Liverpool
University

Group Member (Alphabet Order):

Difeng.Yu

Juntao.Zhu

Liwei.Yin

Xiaonan.Chong

Yao.Bei

Yinlin.Zhao

Website Available On: www.davinyu.com

Abstract

In this software project, our group implemented an online website named “Co-Paper” which provides people an efficient way to organize their reading achievement and make use of collective intelligence. Users can write down the notes about articles or papers in our website and publicize the notes in order to get feedback from others’ comments. This system simplifies the daunting work for typing in all the information for referencing readings and provides the function of using only URL and title to identify each piece of work. We put all our efforts to provide users a simple and interactive literature review system with minimum limitations and proper degree of social functions.

In this report we will provide the detailed explanations about what we have done and how we have done it. Firstly, we will introduce the system design, especially the database design, which is the core of the whole system. Then, the second part will introduce how the whole team members cooperate with each other to carry on the implementation work. This part will concentrate on the development methodology and technique issues we encountered during the implementation process. Thirdly, we present a formal testing report about our website. Finally, more evaluation reports will be listed about system issues, business cases and possible future development.

In summary, our team has implemented the literature review system as we presented in the proposal. The testing shows a positive evaluation about this system. Meanwhile, there still exist much future work for both functional development and research.

Contents

1. Project Description.....	6
● Project Overview	6
● The Purpose of the Project	6
● The Scope of the Work.....	6
● Product Scenarios.....	6
➤ Main Scenario.....	6
● Stakeholders	9
● Mandated Constraints	9
● Naming Conventions and Definitions	9
● Relevant Facts and Assumptions.....	9
2. Requirements	10
● Product Use Cases	10
➤ Use Case Diagrams.....	10
➤ Product Use Case List.....	10
● Functional Requirements	10
● Non-Functional Requirements.....	10
● Activity Diagram	10
● State Machine Diagram.....	11
3. Design	12
● Architecture Design	12
● Database Design.....	13
● User Interface	13
4. System Implementation.....	13
● Development Methodology.....	13
➤ Design process – water fall model.....	13
➤ How to carry on work separately and concurrently	14
➤ Human Resource Allocation.....	15
➤ Communication Plan.....	15
➤ Project Diary.....	17
● How to control and manage the quality	18

● Team Coordinate Technique: GitHub	19
● Official Communication Tool: Slack.....	21
● Programming tools we use.....	21
➤ Back End	21
➤ Front End	22
● Changes.....	22
● Coding Implementation	22
➤ Note Search, Create, Modify,.....	24
➤ Visitor, Member, Premium	24
➤ Reset Password by Email	24
➤ Comments and Reply	24
➤ Dirty Words Filtering.....	24
➤ Page Division	24
5. Test.....	24
● Test Case.....	24
● Alpha Testing – Peer Review.....	24
● Beta Testing	24
6. Project Issues.....	27
● Ethical Issue	27
● Social Issue.....	27
● Legal Issue.....	27
● Business Case.....	27
7. Further Improvement.....	28
● Recommendations for future projects	28
8. Conclusion	29
9. Appendix	29
● User testing instructions and contract.....	29
● User behavior data collection form.....	30
● Weekly Plan 1.0.....	30
● Human resources allocation 1.0.....	31
● Peer review report 1 from backend group member BeiYao	31

1. Project Description

Project Overview

Lots of scholars are dealing with bounds of journal papers and articles every day. There are requirements for recording some notes about these papers, so that readers can recall the contents and key points they appreciate. In addition, people are willing to check other reader's view about the same paper and receive comments targeting his own publicized comments.

Many professors rely on product including "End note", "Mondeley", "Papars" etc. Of course, they have inadequacies. Based on our survey, many people are in desperate need for a literature review system with minimum limitation on the format of the notes. Therefore we proposed to start a project to build this kind of literature review system.

This system simplifies the daunting work for typing in these papers to the system, rather, it provides the function of using only URL and paper name to identify each paper in the servers' database and view it in the local browser. This system also allows all the users to publicize their notes and receive others' comments. Ideally, scholar who does not know each other can form a meaning discussion and exchange views about the same article. We hope this system can help scholars reading papers in a more efficient way.

The Purpose of the Project

The Scope of the Work

Product Scenarios

➤ Main Scenario

Alice is a professor working in Xi'an Jiao-tong Liverpool University and her main research area is software engineering. Every day, she needs to read papers and articles about the latest development in this field. It is often the case that Alice recalls that there is a paper she has read a while ago which seems to provide a great solution for the problem she encounters now, but she cannot recall the exact name of the paper or where did she find it at the first place. It usually takes her a very long time to successfully identify that paper, still, more time is taken to review the whole paper and find the paragraphs she needs to read again. Therefore, Alice decides to use a specific system to help her manage these papers she read. And we recommend her our system.

This is an online system. Alice can use it with her own laptop or smart phone. First she logs into the system, here she can check the hottest notes created and publicized by other users (the notes that receive most comments) and search notes with the key word and general subjects.

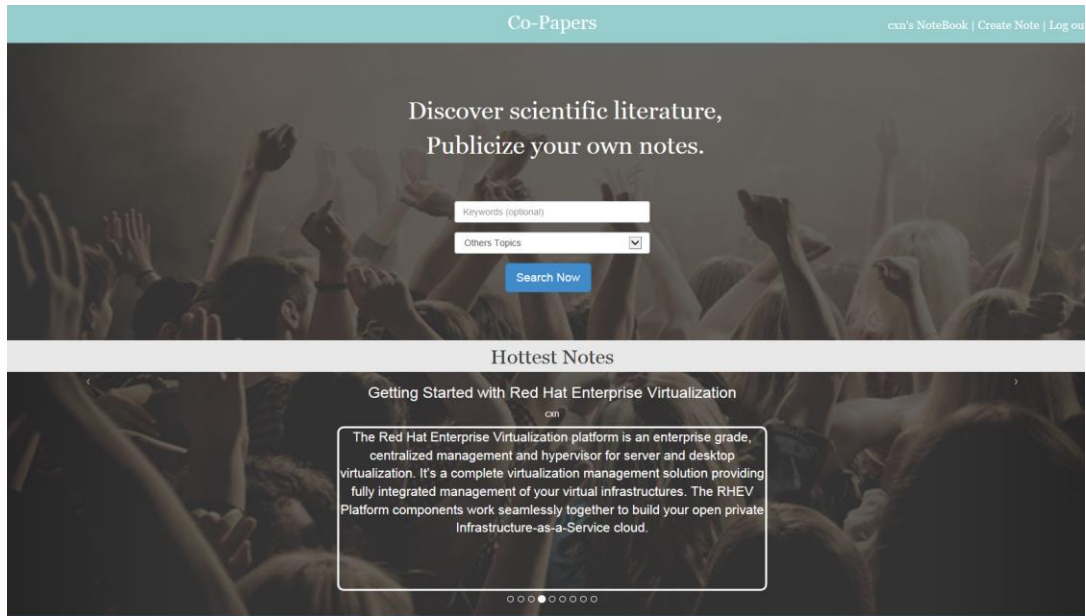


Figure.1 User Enter the Main Page

Click “create new note” button on the top to create your own notes. You can choose to publicize it, which means you allow other users to see your notes or just save it for your own reference.

Figure.2 User is creating Notes

After adding a new note, Alice recalls a paper about how to handle the big team administration and she wants to find that paper and read it again. Click “Hello Alice” on the top to check all of your notes.

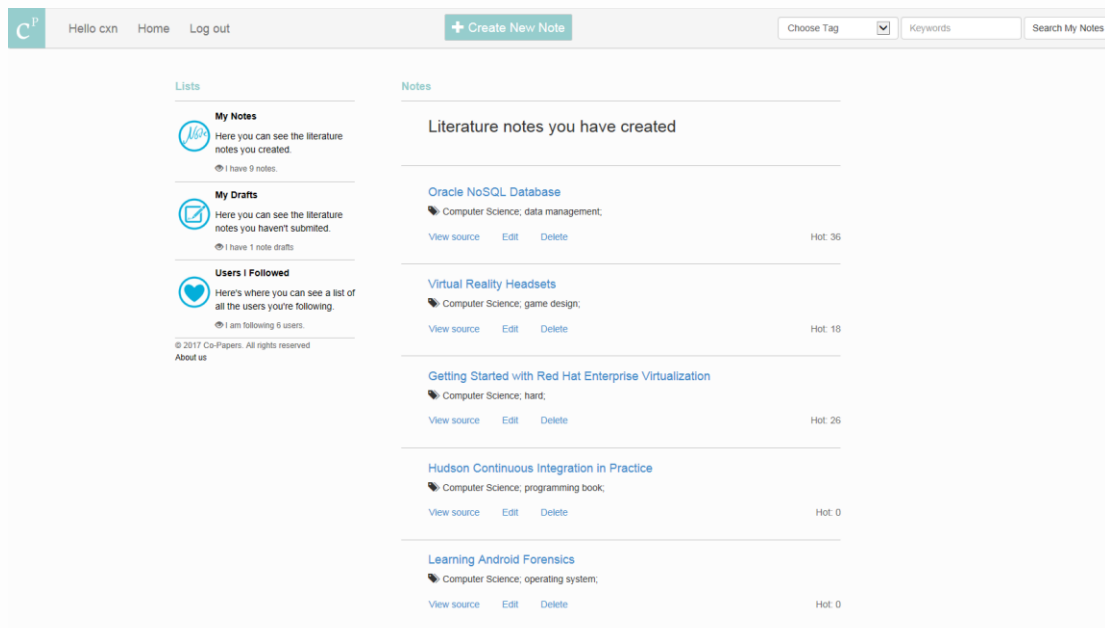


Figure.3 The User Note View

She has created a lot of notes, listed by the time she created the note. It is hard to recall when she created that one she wants. Therefore, Alice made use of the search function and type in “administration” as the key word for search. And she quickly got what she wants.

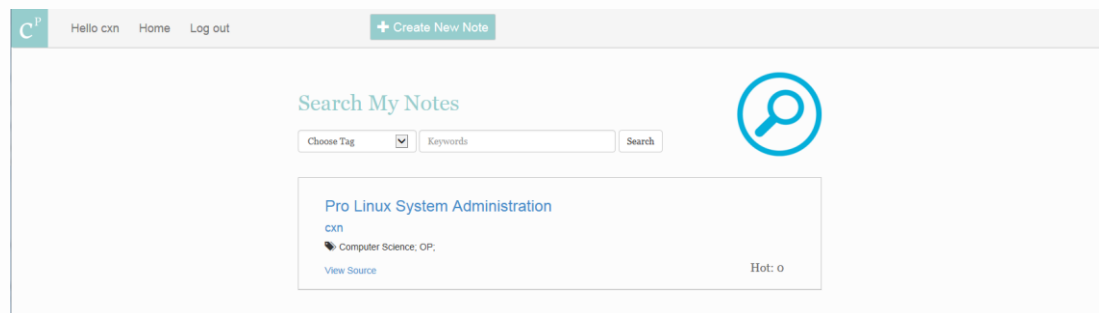


Figure.4 User is searching for his/her own notes

Then, Alice clicked the header to see the details of this note. Here, she can view the source and download the paper again; check the note she left last time, which summarizes the main content of this book; edit this note: this time Alice focus on the third chapter of this book and she can add some sentences about what that chapter mainly about and how can she make use of it in later research; check others' comments and reply them if Alice wants.

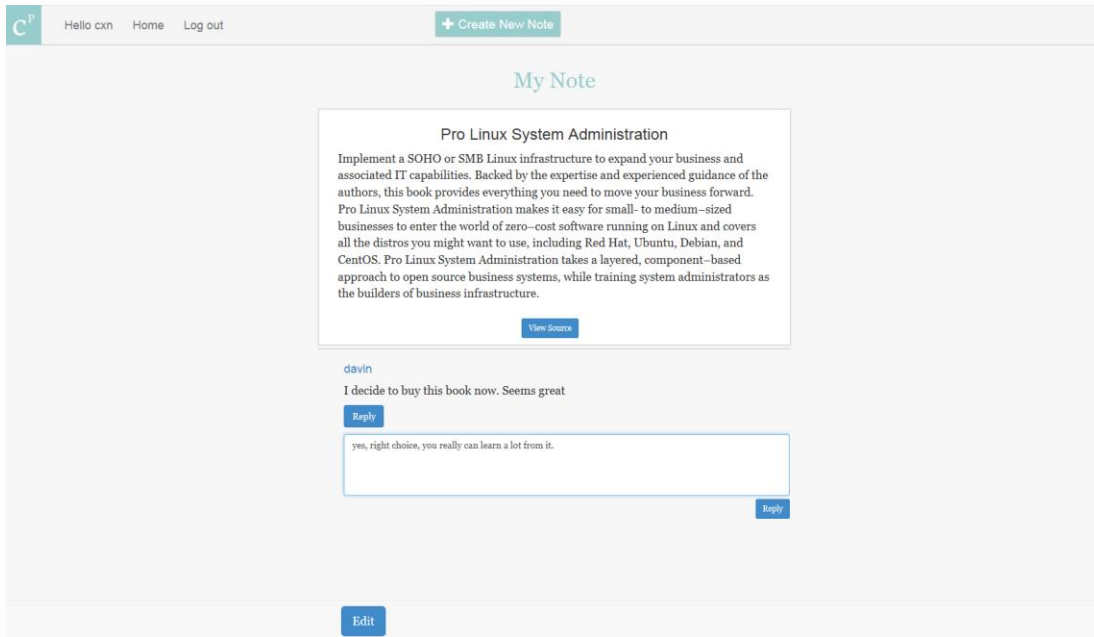


Figure.5 User looking for his/her own note

✚ Stakeholders

✚ Mandated Constraints

✚ Naming Conventions and Definitions

✚ Relevant Facts and Assumptions

We complete the assumption and add some supplements to simplify our system. Besides, we take serious consideration to make sure that these assumptions are reasonable to some extent.

1	One user only needs one tag one topic for one note.
2	One creator only needs to reply once to one comment to finish the conversation.
3	Commenter has no need to reply to the reply from creator.
4	The note which has most comments is defined as the hottest note.
5	Users care more about the creator of one note rather than the note itself.
6	Most of the users are familiar with the terms we use, e.g. "topic", "tag", "note", "comment".

Figure.6 Assumptions

2. Requirements

Product Use Cases

- Use Case Diagrams
- Product Use Case List

Functional Requirements

1	Register and log in to the system.
2	Create note about one article where the user can set user-defined tag and system-made topic to category notes.
3	Save one note or publicize one note.
4	Find your saved notes in “my Drafts” list.
5	Find your publicized notes in “my Notes”.
6	Edit or delete both saved notes and publicized one.
7	Search your own notes by tags.
8	Search others’ notes by topics.
9	Comment other’s notes.
10	Reply others’ comments to your note.
11	Check the hottest notes.
12	Check the creator and his/her note list.
13	Follow other users and put their direct link into your “users I follow” list.

Table.1 List of Main Functions

Non-Functional Requirements

Activity Diagram

The activity diagram shows the graphical representation of an executed set of procedural system activities in this literature review system. It intuitively explains what happens in a workflow, what can be done in parallel and whether there are alternative paths though the workflow.

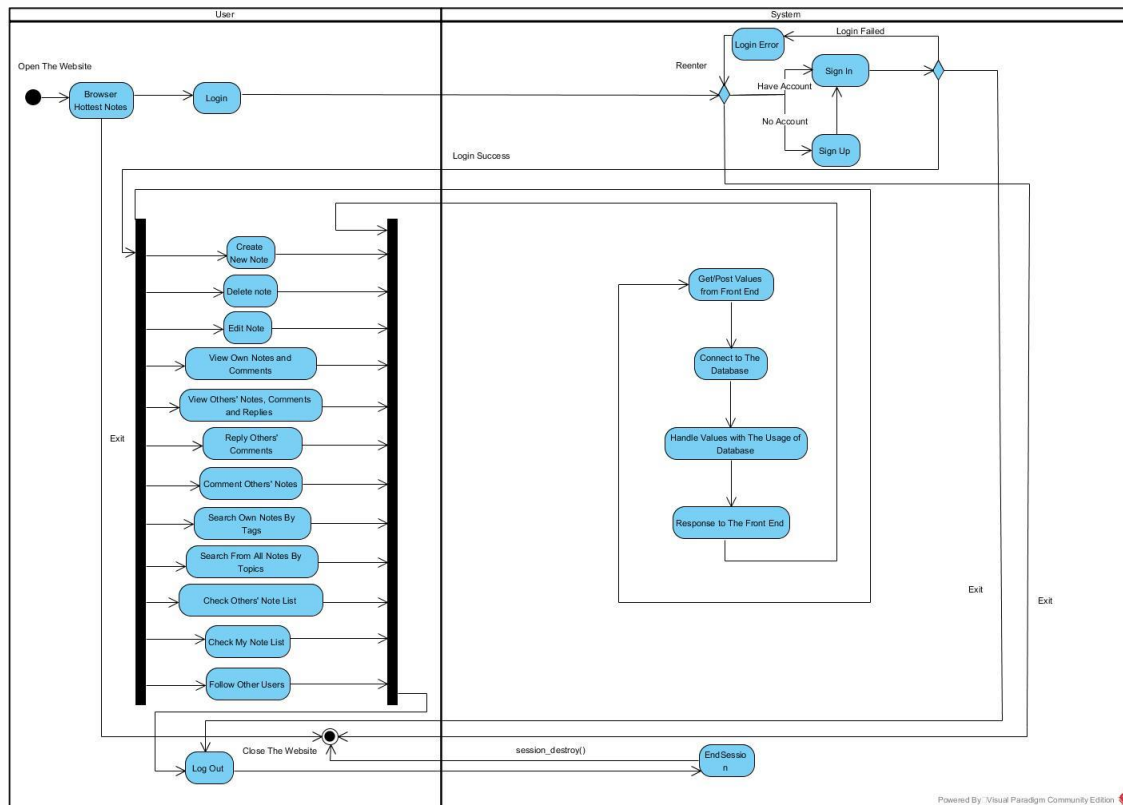


Figure.7 Activity Diagram

State Machine Diagram

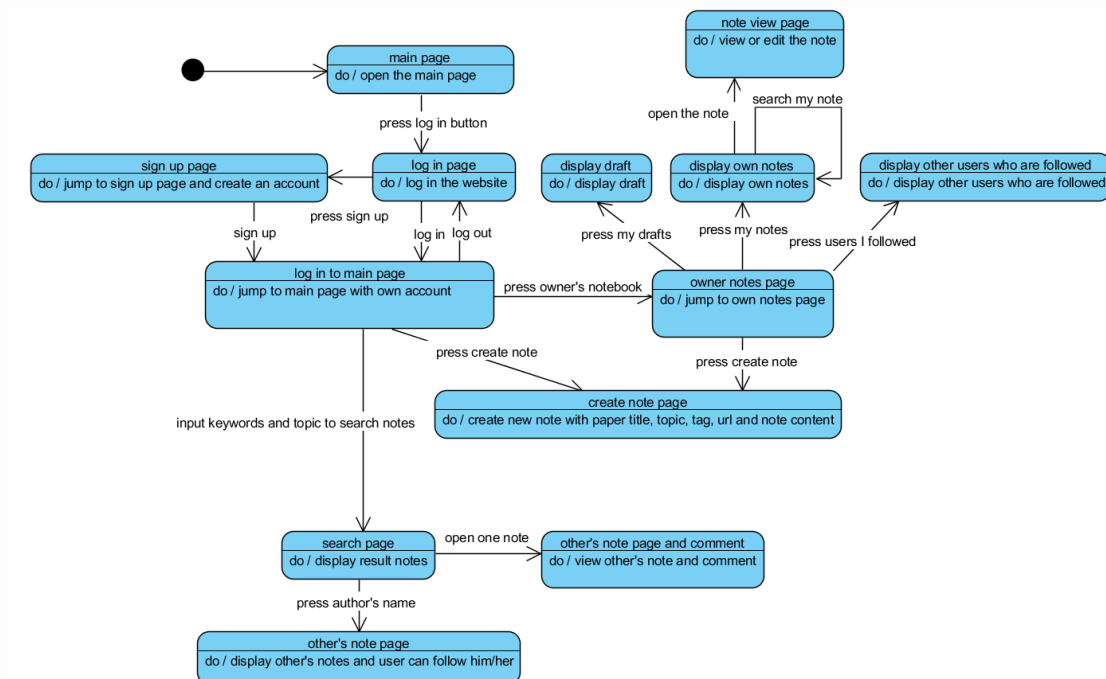


Figure.8 State Machine Diagram

3. Design

In this part and the following section – system implementation, we are going to show how we have implemented this system. The key challenge in our case is how to divide the big project into separate individual works and to allow team members to carry on their piece of work separately and make sure their works do not interfere with each other. Our strategy is to adopt the MVC architecture to explicitly decompose project into three parts and document the interface with everyone's agreement before any one start working on his or her part.

Architecture Design

We adopted the general architecture pattern for web application design to separate components – the Model-View-Controller pattern. More specifically, we use JavaScript, HTML to code the web pages and the transitions between them, which is the view part. We use PHP to code the controller part, which mainly focus on the adding, retrieving, and updating data from database to response the request.

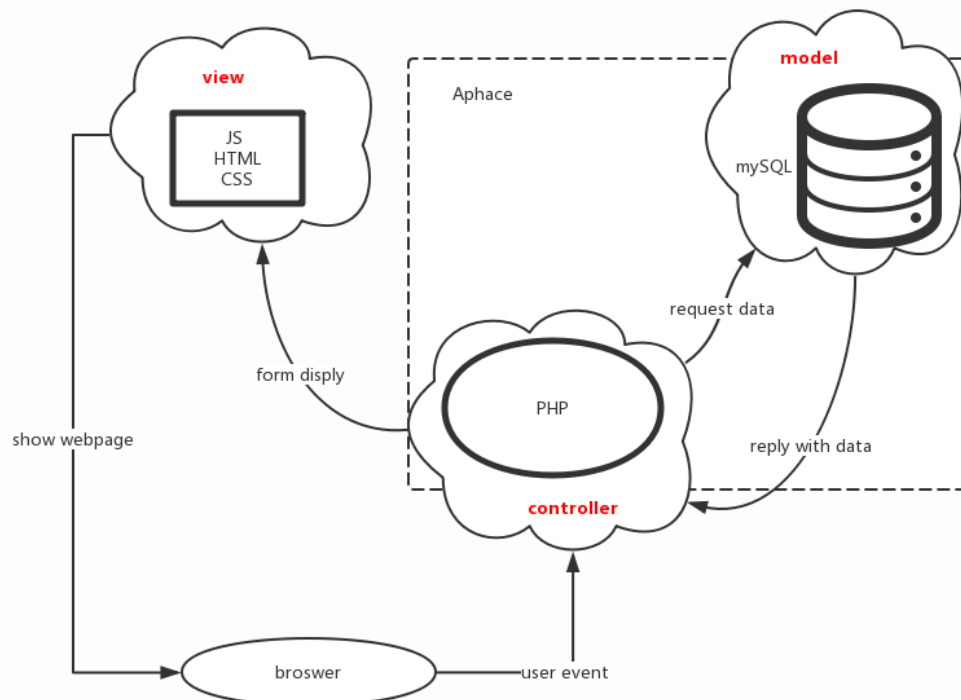


Figure.9 Architecture Pattern

Database Design

The model part is the database we design, where we clarify all the relationships appear in this system. As the introduced concepts presented in the proposal, one user can create many notes, one note can receive many comments, the creator of the note can reply anyone's comments, one user can follow another user, one note can be searched by system defined topics and self-created tags.

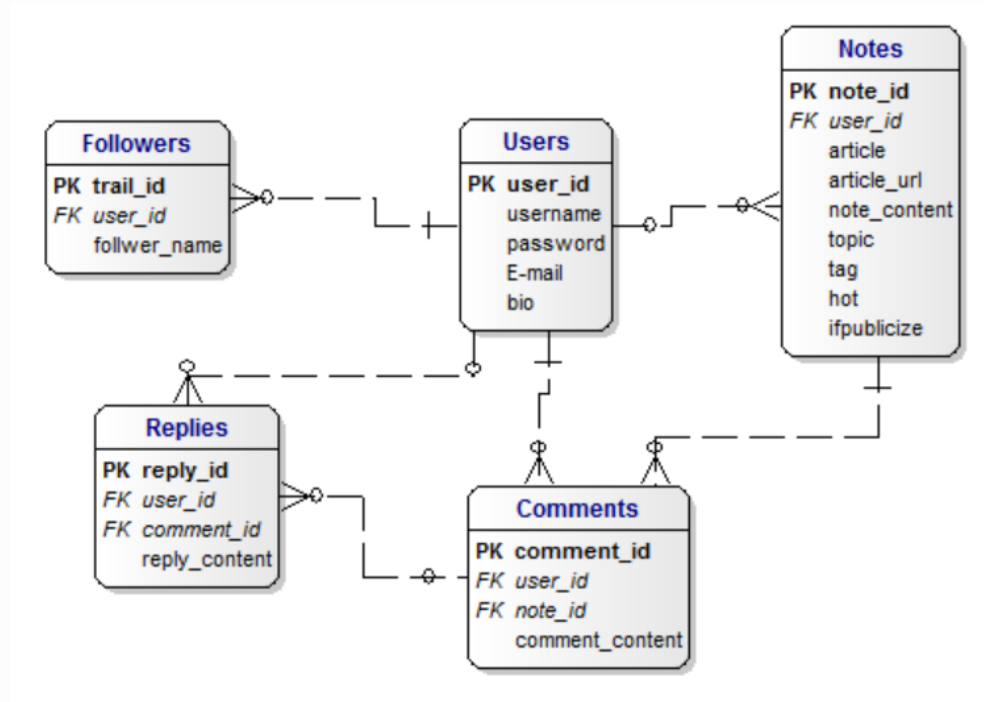


Figure.10 Database Design

User Interface

4. System Implementation

Development Methodology

➤ Design process – water fall model

We stick to the water fall model as we proposed, since the requirement analysis is rather important in this model, we reviewed all the user requirements again after the propose presentation. The following graph shows the actual implementation situation based on our proposed design process.

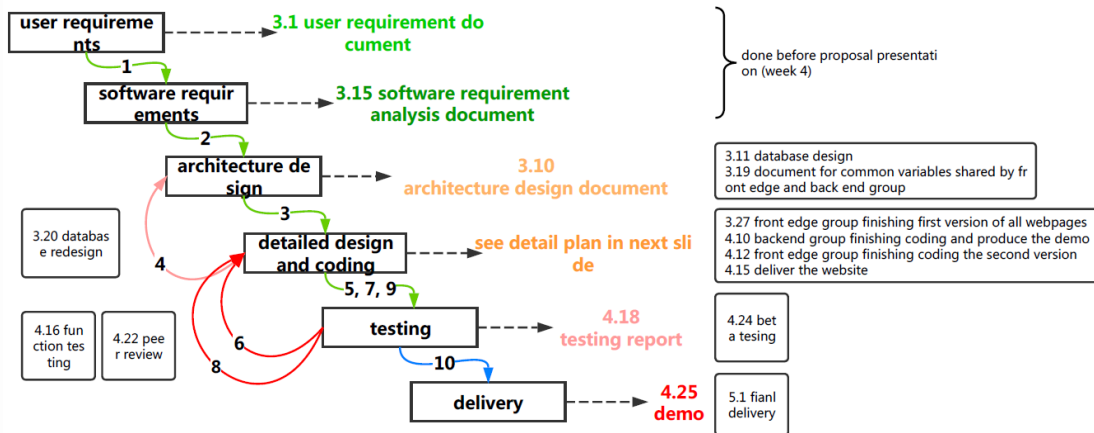


Figure.11 Waterfall Model

➤ How to carry on work separately and concurrently

We follow the human resource allocation plan and weekly working plan as proposed until week 7. We assumed that it should take at least 3 weeks for G2 to get familiar with the developing tools, and it turns out that G2 has been able to start coding at the middle of week 4. Meanwhile G1 finished first version of web pages coding in week 5. Therefore, we adjust the weekly plan from week 8

week	Group 1 front end	Group back end	Group 3 managing
8	Fix some layout problem	Fix the problem in combining three members' previous work	Writing the final report structure
9	Add the "about us" page and "feedback" page	Peer review – review the code; Produce the activity diagram and the state machine diagram	Propose the post-implementation review plan
10	Beta testing – collecting the user feedback		Finishing the first version of final report
11	Post – implementation review meeting		
12			Adding the testing report to the final report
13	Report review		Finishing second version of final report
14			Finishing final version of final report

For detailed weekly plan followed before week7, please refer to the "weekly plan 1.0" in the appendix.

➤ Human Resource Allocation

Also, due to the change of available human resource – one of the team members got car accident and cannot attend school, we reallocate the job. Meanwhile, according to the actual working situation in week 4, we made the **job allocation** more specific.

Human resource allocation starting from week 5

Group	Team member	Main responsibility		Tasks for all
G1 front edge	Juntao Zhu	Coding first version web pages		Working separately and concurrently based on finalized documentation for reference, and report progress and problem weekly.
	Yilin Zhao	Coding final version web pages with proper decoration (logo design)		
G2 back end	Difeng Yu	Coding 4 web pages' back end connection	Establishing same database in individual work place; Review each other's code	
	Yao Bei	Coding 4 web pages' back end connection		
	Liwei Yin	Coding 3 web pages' back end connection; Combining another 2 members' work		
	Jiajie Ni	Car accident – rest		
G3 manager	Xiaonan Chong	Project planning; proposal writing; reporting.		

Table.2 Human Resource Allocation

For detailed human resource allocation followed before week5, please refer to the “human resource allocation 1.0” in the appendix.

➤ Communication Plan

Also we work out the **communication plan** for team members to follow and exchange information systematically

The type of changes	How to communicate	How to handle
System assumptions are found inappropriate by individual	The member raised the problem immediately in the WeChat group	All team members respond in the group, based on the outcome, team manager decide whether to discuss this in next group meeting.
Front edge design and implementing	Members in G1 have a small group meeting	If the change does not affect the overall design, G1 finishing

changes		discussion and report to team manager orally; if it does affect the overall design and G2 implementations, then group meeting should be arranged as soon as possible.
Backend design and implementing changes	Database design changes: G2 in group discussion and report to team manager. Changes on tools and techniques to implement: G2 in group discussion, inform team manager.	If the backend programmers state it is necessary or may bring some convenience to make some changes which will affect the front edge design, G1 shall follow the changes raised by G2.
Note: all the team members can decide whether their decision of making some change will affect others work based on the documentation “common variables settings agreed by all”.		
Changes on time schedule	Team manager raised the change in group meeting	Time schedule will be changed and documented after all members' agreement.
Changes about human resources	Group manager raised to the whole team when making the judgement that some job cannot be finished in time due to limited human resources	Team manager propose a new human resource allocation plan and talk to team members whose job may need to be reallocated.
Changes raised by the stakeholder	Stakeholder raised the change on Monday workshop or problem raised in the testing phase	Discuss in the group meeting based on detailed situation.
Other changes	Raised in WeChat group	Team manager shall keep responsive
Note: team manage document any change if it affects the whole team.		

Table.3 Communication Plan

Considering both the job allocation and the communication protocol, the RACI chart and the RAM matrix are documented to make each member's work more specific.

Activity	Juntao	Yilin	Difeng	Yao	Liw ei	Jiajie	xiaona	KP TA &	Yu tao
Requirement	R	A	C	C			R	C	C
UI functional design	R	R			C	C	I		C

UI design	C	R	I	I	I	I	A	C	
UI implementation	R	R							
Database design	I	I	R	R	R	R	A	C	
Database implementation			R	R	R	R	A	C	C
XAMPP leaning			R	R	R	R		C	
Reporting	I	I	I	I	I	I	R	C	

Table.4 RACI Chart

phase	Juntao	Yilin	Difeng	Yao	Liw ei	Jiajie	xiaonan	KP TA	&	Yu tao
Requirement	p	p	p	p	p	p	p	s		r
Design	p	a	p	a	a	a	p	r		
Development	p	p	p	p	p	p	s	r		i
Testing	a	p	a	p	p	a	s	r		
Training	a	a	a	a	a	a	a	r		p

* P=primary; A=assigned; R=review request; I= input required; S=signature required.

Table.5 RAM Matrix

➤ Project Diary

Week	Progress	Problems	Issues	On schedule?
4	G1 Finishing UI design; G2 Finishing technique investigation – implementing simple database retrieval; Finishing polishing proposal presentation.	Allocated and negotiate detailed job for members in G1 and G2 - DONE	Proposal presentation	YES
5	G1 finishing coding 6 of 9 first version web pages; G2 finishing coding the log in and register web page; Finishing redesigning the database schema.	Determine the finalized version of document about the common variables shared by G1 and G2 for reference – DONE;		YES

		Privacy issue – CONTINUE BY G2;		
6	G1 finishing coding first version web pages; G2 finishing coding the 6 of 12 backend connections;	Determine the final version UI design – DONE.		YES
7	G1 finishing decorating 4 of the 12 second version web pages; G2 finishing coding all the backend connections;	G2 raise problems about JS and JQuery – CONTINUE.		AHEAD
8	G2 finishing merging all the links between webpages and present the demo. Whole group review the website and find places which need to be improved	Test problem – 2 of G2 start code review; Need to reschedule the weekly plan - DONE	First version demo	NO: "my note view" page not done
9	Peer review – code review User test – counting the average time the users takes to successfully create a new not	There are problem with the searching function: input cannot be empty. We want to allow users to check all notes in one category.	Group report first draft	AHEAD
10	Discuss the user testing outcomes	Continue fix the reply comment function	Testing report	YES
11			Final version demo	
12				
13			Final report	
14				

Table.6 Project Diary

How to control and manage the quality

We have finished web page design with mock up and database design in week 3, all the progress before week 4 are fundamental and essential, thus are reviewed and agreed by all the members in the weekly meeting. The following picture shows the quality control plan since week 4, after works are done separately.

For G1 working on the web page design and coding, two group members working in parallel and review each other's code.

For G2 working on the database design and implementation, Difeng Yu is in charge of combining all the G2 members' work, thus, another members report to him and get feedback.

For G3 working on documentation, the only member do self-review and post all formalized document in Slack for team members to reference and review at any time.

All groups report their progress every week and review each group's work and provide feedback for improvement.

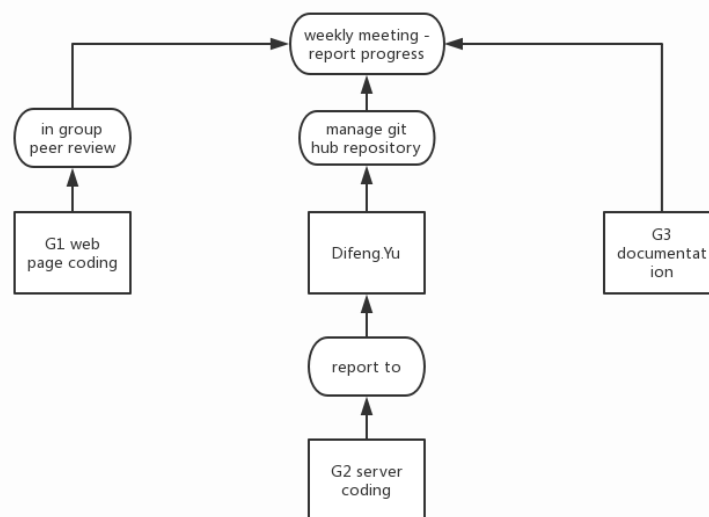


Figure.12 Management Structure

Team Coordinate Technique: GitHub

According to our team structure, which makes the front edge and back edge separately, the version control system is a must use choice for our code to accomplish the website simultaneously. We choose GitHub to manage our codes and we made a private repository on the website. As we started our development, we first pick two leaders, one from the back edge and one from the front edge, to push the commits to the GitHub in the development branch. After the work the checked by the master, he will merge the development branch to the master branch to form a new release. In this condition, the other members of the team can always get the newest version of our project and continue his/her work. If one team member is not satisfied with some part of the current work, he/she can issue it on the GitHub.

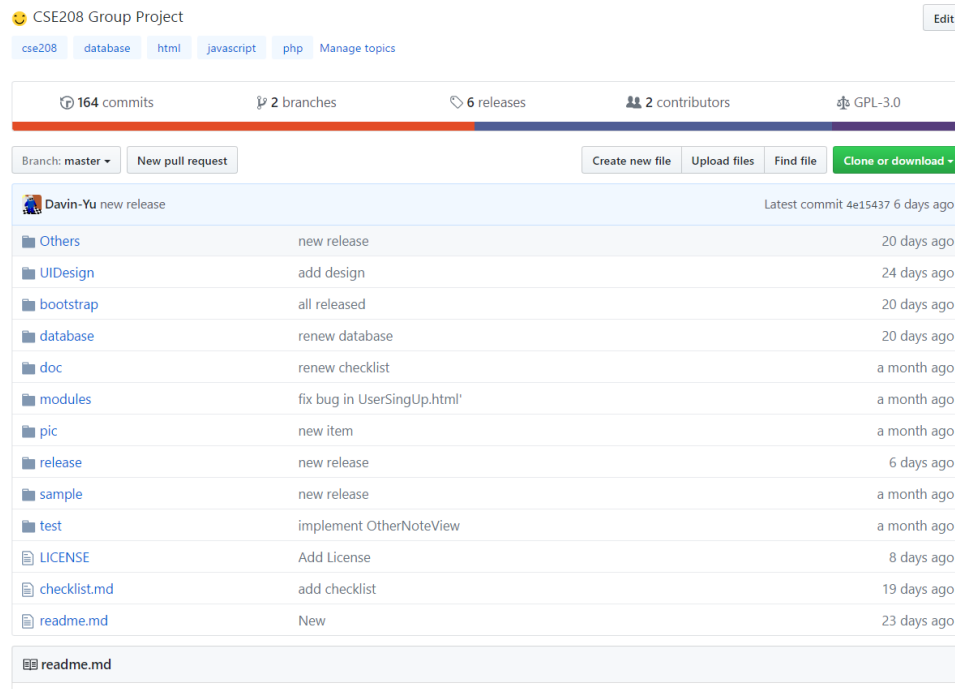


Figure.13 GitHub Repository Main Page

During the development, the coordinate (which is G3 manager) can always check if the development team finishes their work at a specific time or not. She can also check the code frequency and commits graph easily, in order to prevent the potential accident from happening. For example, as our code frequency diagram shows below, the G3 manager can easily figure out what is the peak time of our coding process, is the code robust or not, is there any refactoring of the code needed.

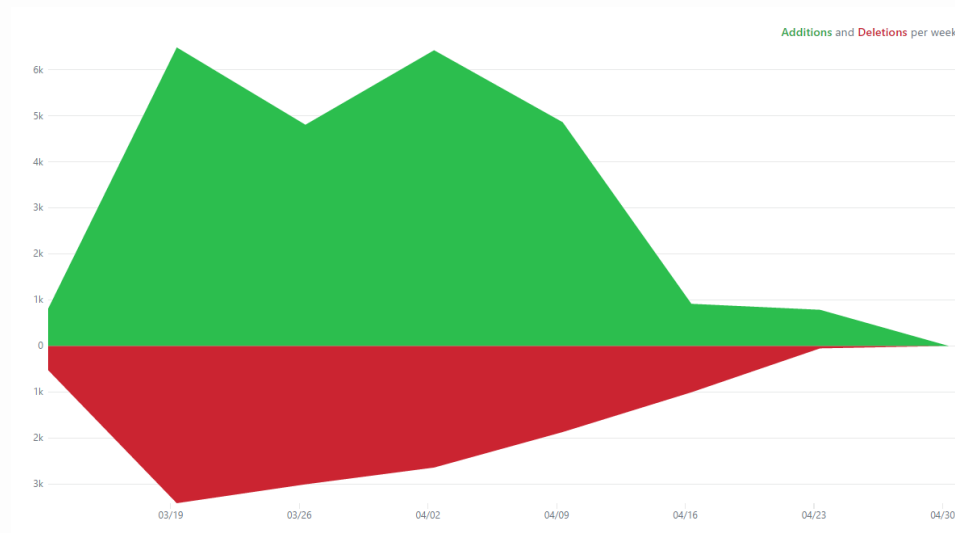


Figure.14 GitHub Code Frequency Figure

Official Communication Tool: Slack

Unlike our daily communication in WeChat, we use Slack as an official communication tool during the project time to improve our efficiency. We found that talk through WeChat is easily to miss some important files and difficult to find the previous files. We use Slack thus, as a tool, to launch and save important files which are mainly created by G3 manager.

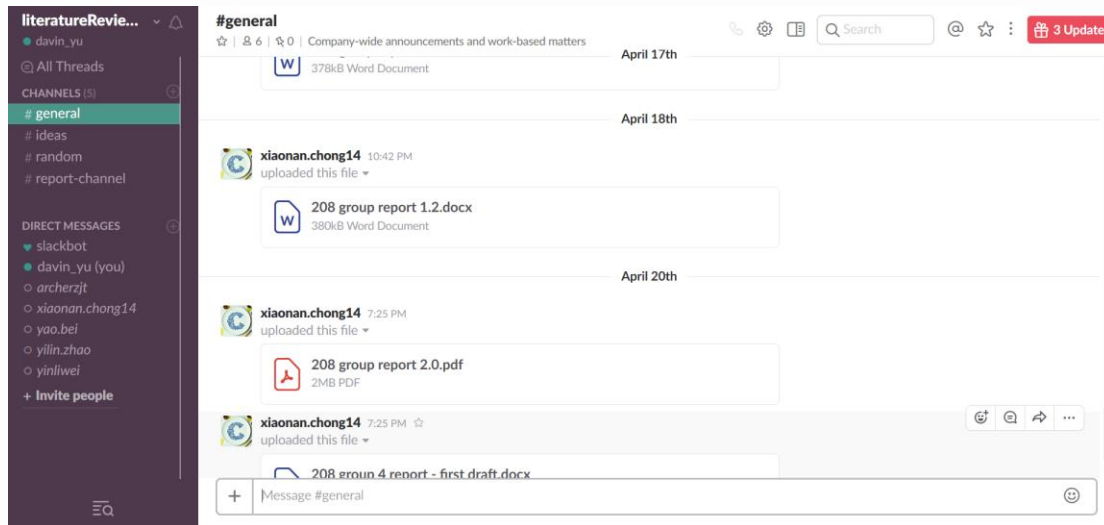


Figure.15 Slack Screenshot

Programming tools we use

➤ Back End

There are a lot of tools well developed for coding a web site. We started with none of our team members knowing exactly how to use any one of these tools. Based on our investigation, about coding language, there are roughly 30% people using PHP; 60% using JAVA EE and 10% using nodeJS. It turns out that nodeJS **allows** your application to deal with unlimited requests because it uses one thread and applies scheduling algorithm. However, it is rather complex for us to learn in 2 weeks and to use for coding in the rest 4 week. JAVA EE has stable and enterprise level application such as Tomcat to support; Meanwhile, PHP is developed much earlier, thus, there tends to be more books and free online guidance.

We run a simple test to find out which tool is more convenient to use and which one is easier for this team to get familiar with. We set one group to learn PHP and one group JAVA EE, after one weekend, both group take a simple test – writing a telephone book for with only two functions: inserting a pair of information and retrieve one pair. PHP group takes much less time to implement the test. We then decide all the back end group will code with PHP.

➤ Front End

At the beginning of the coding, we planned to implement the UI design without using any web frameworks. However, after realizing several static webpages by primitive HTML and CSS, we found that it is difficult to realize the responsive webpage which is an important requirement of front edge. Otherwise, it consumed a lot of time for our coders to implement the complicated UI which needs a lot of JavaScript.

By the recommendation of tutorial assistant, we determined to use Bootstrap framework to improve the efficiency of front edge building. Bootstrap is a front-end framework that can be developed with JavaScript, HTML and CSS. The ready-made coding blocks offered by Bootstrap make it easy for us to build the website without coding from the scratch. The Grid Style of Bootstrap helps us to realize the responsive webpage easily, and the Base Style and the Pre-styled Components of it helps us implement the complex UI such as foldable responsive navigation bar and dynamic comments updating box.

Changes

Considering the typesetting layout, we remove the “newest notes added by the users you are following” in the main page. Except that, we have implemented all other functions included in the first proposal and add some new functions to provide a more user friendly system.

1	Allow the users to choose whether to publicize one note, each user can save his/her notes only for himself/herself referencing. And the saved notes can only be searched though “searching your own notes”, and cannot be found though search box in the main page.
2	Delete notes (including notes that are saved or publicized).
3	Provide a “about us” web page for users to know more about us and the system.
4	Provide a “feedback” button in the main page for users to leave any messages about how to improve the web site and their further requirements.

Figure.16 Changes in the system

Coding Implementation

As we began to implement all the functions for the website, we first divided different pages into different folders. Each folder represents a single webpage which contains the related HTML file, CSS file, php file and PNG/JPEG using in the current webpage. For example in the ‘MainPage’ folder, we will have: ‘MainPage.css’ for assign CSS style to the main page, ‘MainPage.php’ file which includes the HTML and PHP call functions for the main page display; We will also have ‘getHotNote.php’ file which can be called from ‘MainPage.php’ to get the hot note from the database; Other files such as background

picture is also contained in the folder. By doing this, we will not only have a clear idea of which function is used for what, but also it makes our refactoring to the code easier.

AboutUs	change font in aboutUS page	11 days ago
CheckDirtyWords	check dirty words	4 days ago
Feedback	add feedback	17 days ago
FindPassword	add email reset password	4 days ago
Login_Main	Computer Science as the default search	4 days ago
MainPage	Computer Science as the default search	4 days ago
MyNoteEdit	check dirty words	4 days ago
MyNoteView	repare a bug	4 days ago
MyView	user friendly	11 days ago
MyViewSearchedResult	user friendly	11 days ago
NoteCreate	add premium	4 days ago
OtherNoteView	user friendly	11 days ago
OtherView	add "divide page"	4 days ago
Premium	add premium	4 days ago
RandomResult	user friendly	11 days ago
SearchedResult	user friendly	11 days ago
Uploadpdf	enable pdf upload	4 days ago
UserLogin	new release	17 days ago
UserSignUp	new release	16 days ago
connectDB.php	to local	20 days ago
testLogin.php	new release	24 days ago

Figure.17 *GitHub Release Version*

For the front end and back end cooperation, we first unified the variable names which are used for both side. We implement small functions in php file which can be called from the main PHP file which contains the mainly of the HTML. This makes our work division easier and everyone can be responsible for the part he/she wrote.

We will then explain some sample code for different functions:

- **Note Search, Create, Modify,**
- **Visitor, Member, Premium**
- **Reset Password by Email**
- **Comments and Reply**
- **Dirty Words Filtering**
- **Page Division**

5. Test

Test Case

Alpha Testing – Peer Review

We started our testing process with peer review; the mean problems are summarized as the following mainly focusing on the comments.

1	Shortage of functional description in the comment section
2	Several parameters have not been claimed in the comments
3	In the code section, declaration of variables' types have not been commented

Table.7 Peer Review

For detailed information and explanation, *please refer to the “Testing: peer review report 1 from one of our backend group member in the appendix.*

Beta Testing

About the how to calculate the hottest notes, we made an assumption and define hottest notes as the notes with the largest number of comments. There is concern that notes listed in the home page as hottest notes are tend to receive more comments, thus more possible to remain as the hottest note. However, in this specific environment, we take scholars studying specific fields as our target customers, therefore it may not be the case that one user is able to comment notes about articles outside his major.

To release our concern, we designed a special purpose test to examine the assumption – “The hottest note is the notes with the largest number of comments”. We release our website for user testing and observe the average time for the hottest note shown on the home page to be replaced. We check the website at 12am every day and record whether the hottest on remain the same, if yes, record 0, if no, record 1. We collect the number from week 10 to week 12. And the data we collected shows that

Week 10		Week 11		Week 12	
4.24	1	5.1		5.8	

4.25	0	5.2		5.9	
4.26	0	5.3		5.10	
4.27	1	5.4		5.11	
4.28	0	5.5		5.12	
4.29	1	5.6		5.13	
4.30	0	5.7		5.14	

For the non-functional requirements, we do some tests and collect the data about users' behaviors. And the summarized results compared with the non-functional requirements we present in the proposal is listed below.

Please refer to the "User testing instructions and contract" and "The user behavior data collection form" in the appendix for how we carry on the test, what kind of information we gave to testers and how we collected the data.

The result shows that the average time random users use to figure out how to create a new note is 3.95 minutes. And the average time to successfully follow another user is 4.26 minutes.

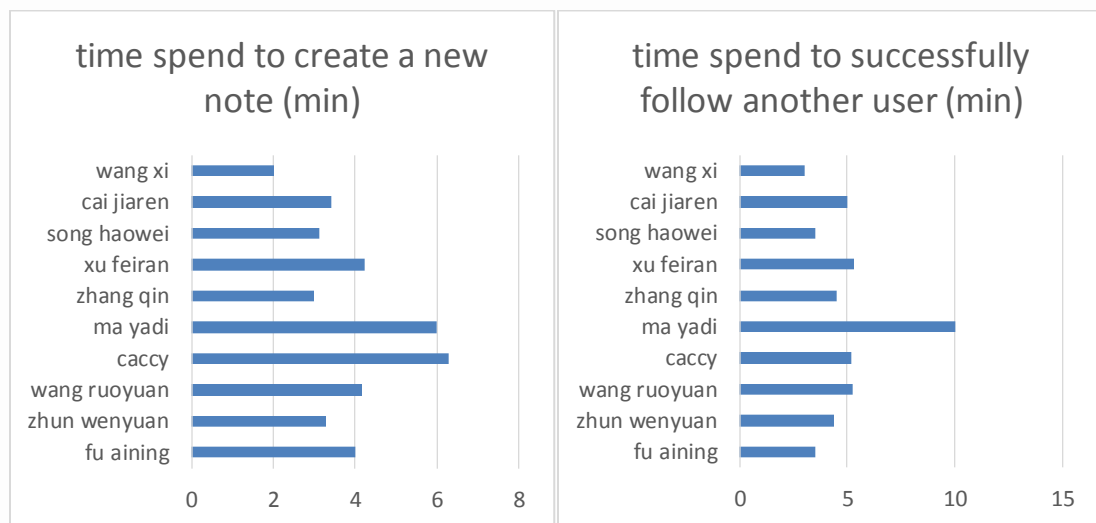


Figure.18 Time Chart

Non-functional requirements	Accessible criteria	Result
Response time	The website response each request within 1 second.	In XJTLU campus, the website response each request no more than 2 seconds.
Reliability (Availability)	The website can handle more than 50 requests concurrently.	No problem
Maintainability	Provide well-organized document	

	which can be understood by any programmer outside the team within 2 hours.	
Usability	User can learn how to create a new note in 5 minutes; and can learn how to comment and follow others in 10 minutes.	The average time to create a new note is 3.95 minutes, the average time to successfully follow another user is 4.26.
Robustness	Catch all the exceptions: whenever there is an error, the system provides detailed instructions and solutions which guarantee that the user can fix the problem in 15 minutes.	
Privacy (Security)	No intruder can get user's password.	The password is encrypted and even the developer cannot access the password.

Table.8 Test for non-functional requirement

Review	How	Who
If functional requirements are met	Check whether the elements of websites are complete based on the proposal	G1
If non-functional requirements are met	Check the response time, robustness, stability by alpha testing; beta testing and survey on user feedback	G2
If all the work is documented properly	Invite computer industrial peers review all the documentation to see whether it states clearly	G3
If all the work is done in time	Compare plan and actual implementation record	All team & stakeholder
Peer review: if everyone done his job properly	Having final team meeting and have a talk;	All team
Possible future improvement	Finishing individual report;	
Problem remained unsolved	Each group report problems unsolved.	
Asking for suggestions from tutor	Receive suggestions after final report	All team & KP, Sami

Table.9 Post-implementation review before closing the project

In conclusion, this website has fulfilled all the requirements mentioned in the proposal.

6. Project Issues

Ethical Issue

This project will not hurt anyone's benefit: 1. this is a free website; 2. other websites shall be willing to see the development because we use their website link to identify each paper which can be seen as a kind of advertisement.

Social Issue

We believe that we are following the trend that in this era of knowledge fragmentation, people need to consider more about how to organize their reading achievements and knowledge well. We want to sell the belief that finish reading is not the only thing that matters, more important thing is to digest and recall past reading often.

Legal Issue

It will be more convenient if the system allow users to upload papers in PDF format with annotations on them. However, we also use the link to identify one particular paper and people will share the link. Therefore, someone created this note may have pay for some database and have the right to download it while others do not. So, we define the URL as the identification where you find this paper.

Business Case

Here we propose several possible ways for the system to make profits:

- **Adding advertisements about new published books or articles:** The publisher may need online ways to let their books be aware, since more people prefer electronic books to paper books and people are less possible to go to book store to find a book they want recently.
- **Adding advertisements about other website whose main function is searching for desired papers:** Many websites provide the function for users to search and down load some book with certain payment. This kind of website may be happy to advertise on our website where people tend to check some books or papers if they find them interesting.
- **Charging for certain functions:** The main function of our website is managing users' notes; while the social and cooperative functions can serve as advanced functions and need uses to pay a little money. For example, the "follow other users" can serve an advanced function.

Achievements compared to business case objectives:

There are mainly two objectives needed to be reached for business use: the first is stability and the second is convenience. We have established a complete system and it is put online for normal use. Therefore there is large possibility for us to find advertisement investments if we keep running the website and maintain the user activation. The first objective is achieved. As for charging for certain functions, the social functions we provide are rather limited and not fully developed. Thus, we consider that

there is still a long way for us to make money in this way. The second objective is not fully achieved.

7. Further Improvement

There is still large room for improvement. We consider the following 3 catalogs as the main possible directions.

	Miss behaviors and exceptions should be further considered to guarantee users can explore the website smoothly.
1	Retrieving the password – it should be possible for user to retrieve his/her password by email confirmation if he/she forgets the password.
	Provide more convenience for users and to the functions rich of the system.
2	Add more than one tag and topic for one note.
3	Check the notes (both one's own notes and others' notes) categorized by the topics or tags.
4	Inform user that at least one of the people they follow has updated their note list (by showing the new note or showing the number of updates).
5	Design a better algorithm making use of more meaningful information to calculate the hottest notes.
6	Provide recommending functions – show some notes which the user may have interest in.
	The current version keep managing one's own notes as the key function, however with the increase importance of cooperation and information exchange, the website can provide more social functions.
7	Allow users to import other's note and do minor change and, meanwhile, the system should automatically acknowledge the prior creator.
8	Introduce the concept – “shared notes” which gather all user's intelligence to produce some kind of formal notes which can serve as the Wikipedia for an article.

Table.10

Recommendations for future projects

Besides the possible functions listed ahead, more research can be carried out about the hottest notes calculation. Because our original purpose is to provide user some examples of the best notes, it is more natural to define the hottest notes as the notes that received most positive comments. For later research, we can further explore semantic recognition techniques to let machine automatically distinguish between positive and negative comments attached on the notes and calculate the “hottest” notes. Knowledges and techniques about machine learning shall also be adopted.

Moreover, search algorithm can be further developed to provide the users with the content they truly want. For example, two closely related notes may not have the same

key words in their titles or in their note bodies; instead, they may share same reference in the articles these two notes targeting to. Many search engines are not able to extract these kinds of related notes for the user and there is still large room for progress. Difficulties shall be seriously considered including how much information should be checked; how to represent the information and the relationships between them; how to store and extract these information.

8. Conclusion

In conclusion, we have not only implemented all the functions we have proposed, but also add several functions for users' convenience. Even though, there are certain limitations constrained by our assumptions which aim to simplify our system, more features can be developed to fulfill the system if more time is given. We conclude that all the assumptions are made properly which allows us to implement all of them with time limitation.

Meanwhile, our team shows excellent cooperation between members, six of the seven members have fully contributed to this project. There is no delay in work submitting and buck passing. Everyone shows good understanding of his or her responsibilities and made best effort to solve the problems they encounter, showing the high ability to fully take charge of one part of work individually.

If we are allowed to do such a project again as a group, we would be bolder and aim for a more realistic system with less assumptions. When facing the uncertainty, we finally decided to take the conservative solution and underestimate our potential. However, next time, with the mature understand about establishing a website and software development process, we certainly can present better performance.

9. Appendix

User testing instructions and contract

Website testing instructions and contract

First, thank you for helping us with the user test for our website – CoPaper.

The web address is: www.Davinyu.com

Instruction:

Please open the website on your own computer and try to get familiar with the all functions with the aim of making use of this website for you own benefit.

We will watch your behavior on your side and do some record. We promise all the data collected will only be used for scientific research and will not be abused.

Contract:
☐

I agree to join this test with my behavior being recorded.

Sign _____

Date _____

User behavior data collection form

Website testing – data collection form

The tester's name: _____

Age: _____

major: _____

the time he/she used to create a new note successfully: _____(min)

the time he/she used to successfully follow another user: _____(min)

any other feedback:

(skip this part if the user has found the “feedback” button and leave some message there)

Date _____

Note: no more information shall be given to the tester while testing.

Weekly Plan 1.0

Week	Group 1 front end	Group 2 back end	Group 3 managing
1	Brain storm		
2	Requirement engineering		
3	Dig deeper into the project; Unify the understanding; Focusing on the realism analysis.		Produce detailed plan; write proposal
4	Finish UI functional design;	Get familiar with PHP; Implementing basic function: log and register.	
5	Provide course pages with no decorations.	Do a simple presentation about how does it work; Decide the detailed division of task in the	Document the variable left unhandled.

		group meeting.	
6	UI design – produce more than 2 versions of page design	Based on course pages, start coding; Finish building the database.	Writing deliverable document on UI design for customer.
7	Confirm with customer.		Confirm with customer.
8	Get familiar with the framework.	Finish coding	
9	Finish implementing the well-designed web pages.	Focusing on privacy handling.	Document current system functions
10			
11	Testing each assessable criteria.		Document final version.
12	Beta testing; Deliver software to customer.		Start final report writing.

Human resources allocation 1.0

Group	Team member	Main responsibility		Tasks for all
G1 front edge	Juntao zhu	Implementing 4 of the 7 pages		Requirement analysis; UI functional design; Attend the weekly meeting and reporting personal attribution.
	Yilin zhao	Implementing 3 of the 7 pages; Contacting G2.		
G2 back end	Difeng yu	Implementing request handling functions. log in function; Register function; Create notes; Tag; Comment others' notes; Following interesting people;	Database design.	
	Yao bei			
	Liwei yin			
	Jiajie ni			
G3 manager	Xiaonan chong	Project planning; proposal writing; reporting.		

Peer review report 1 from backend group member BeiYao

There has been a shortage of functional descriptions in the comment section, which tends to cause other developers' confusion of improving the code and further resulting in difficulties on maintainability of the website. So more descriptions of classes and functions should be added for further developments. The following is an example part where functional descriptions are required.

```
<?php
    $topic = $_POST["_topic"];
    $keyword = $_POST["_keyword"];
    Echo
    location.href='./SearchedResult/SearchedResult.php?_topic=$topic&_keyword=$keyword'
</script>";
?>
```

Several parameters have not been claimed in the comments, which should be further improved for the modifiability of code. For instance, *@param* is required to indicate parameters for the following code.

```
/**
 *      @author Davin Yu, Yao Bei
 */
$user_id = $_SESSION['user_id'];
$article = $_POST['_article'];
$topic = $_POST['_topic'];
$article_url=$_POST['_article_url'];
$note_content=$_POST['_note_content'];
$ifpublicize=$_POST['_ifpublicize'];
```

In the code section, declaration of variables' types has not been commented, which requires further complementary. Taking the code in 'getHotNote' as an example, the types of variables 'now' and 'first' have not been declared in the comments which are actually expected to be indicated by *@var*.

```
$now = 0;
$first = 'active';
```

Besides, there have not been declarations of information contained in the included specified files which should be added for improvements of understandability.

```
<?php session_start();
require("inchot.php"); ?>
```

Distinct code visions have been provided. However, there has no revision history contained so that it is quite hard to achieve traceability and maintainability of code.