Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 510 – 519

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

CodeR: Real-time Code Editor Application for Collaborative Programming

Aditya Kurniawan[1]*, Aditya Kurniawan[2], Christine Soesanto[3], Joe Erik Carla Wijaya[4]

[1234]*Bina Nusantara University, JL. KH Syahdan No. 9 Palmerah, Jakarta Barat 11480, Indonesia*

Abstract

The world of Internet is growing rapidly, many applications that previously created on the desktop start moving to the web. Many applications could be accessed anytime and anywhere easily using Internet. Developers need tools to create their applications, one of them named code editor. The purpose of this research is to design and develop a real-time code editor application using web socket technology to help users collaborate while working on the project. This application provides a feature where users can collaborate on a project in real-time. The authors using analysis methodology which conducting on a study of the current code editor applications, distributing questionnaires and conducting on literature study. CodeR is a web application that provides workspace to writing, perform, display the results of the code through the terminal, and collaborate with other users in real-time. The application main features are providing workspace to make, execute and build the source code, real-time collaboration, chat, and build the terminal. This application supports C, C++, and Java programming languages.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Keywords: Code Editor, Real-time, Web Socket, Application, Collaborative Programming

1. Introduction

The technological development trend in software engineering has been improving, where the design of software began move from the desktop to the web. Nowadays, many IDE (Integrated Development Environment) applications

* Corresponding author. Tel.: +62-21-5345830
E-mail address: adkurniawan@binus.edu

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.531

has been made, such as Eclipse, Visual Studio, etc.[1], but IDEs which based on desktop still have significant disadvantages such as long time for configuration and installing the plug-in needed for IDE to run the project. This problem could be a huge waste of time when there are many devices that have to be configured[2].

Many software applications have been run in the cloud, and use a web browser as a user interface that allows ubiquitous access, instant collaboration, and avoid installation and configuration on desktop computers[2].

One of the technologies used for instant collaboration is single IDE (like pair programming). Pair programming is the practice of having two programmers access and work on the same code in a single development environment[3]. In pair programming, programmers have the abilities to create, edit and delete source code in real-time. Pair programming could solve the synchronization problem of program code in order to remain valid, and whenever the code changes any programmer who is working on the same project could see the one who changed the code.

Collaborative technologies could help programmers work together while fixing bugs or discuss the program in the same single environment but in different geographical area. Therefore, it needs to make an application that can improve performance while creating program such as real-time collaboration, create, execute and display the result of the program using terminal.

1.1. Real-time Communication

Real-time Communication (RTC) is merging of communication and collaboration systems, which combined of communication technologies, like Voice-over-IP (VoIP) telephony and instant messaging, and various collaborative application[4,5]. RTC technologies consist of four interconnected building blocks; consist of unified communications, presence awareness, contextualization and E-Collaboration portfolio[6].

Table 1. Building Blocks of real-time collaboration systems[6]

Unified Communication (UC)	Presence Information	Contextualization	E-Collaboration Portfolio
Various media and communication channels	Signalling of people and communicatioı channels	Integration witł office software anc enterprise applications	Audio and video conferences, web seminars
Media and device integration	Aggregation of presence information on group, role, and objective level	Context specific buddy lists	Ad hoc application sharing
Rule-based configuration of message routing and call diversion			Joint whiteboards and discussion forums
Definition of preferred media	Active management of buddy list	Mobile RTC with location-based services	Team calendars and contact management
Unified messaging portal	Individualized and automatic signalling		Document folders

RTC systems usually enable two-person communication and support multi-person conference. By providing and integrating a range of synchronous communication media in one integrated environment, RTC systems allow users to collaborate in real-time, for example editing a document, voice call, multi-person video conference. RTC technology could help to solve the synchronization problem, especially when working in a team. RTC systems offer flexibility and interacting remotely with other users, also it has been facilitated and improved in terms of interaction and communication that could help construct the project more efficiently.

1.2. State of the Art

State of the art of this research is to design a code editor application which has the ability to do a collaboration while working on specific file, syntax checking, run and build those source code through terminal, and users could communicate with the others using chat as media. This application provides workspace where users can create, edit, run and build source code which has been written before and some useful feature like auto complete in C, C++, and Java. For users who want to export the source code files to desktop or import project files from desktop could use upload and download features. Hopefully, this application could be able to help programmer to do some project collaborate in real-time with others, and increase project development performance.

2. Related Research

2.1. Real-time Collaborative Programming

While working on development projects, any programmers working on the project by team. Any programmer who has the access to the project can create, change, and add code inside the same project file. So synchronization process is required between programmers to avoid code duplication, and to solve this synchronization problem integrated real-time collaboration is needed in a single environment.

The Integrated Development Environment (IDE) is focused to provide collaborative setting for programming teams which has the ability to do real-time text editing, run and build code, chat, and various other features. The ability for editing text in real-time allows multiple users to work together while editing a document and display the changes directly to other users who has the ability to access the same document. There are a number of free applications that support real-time text editing feature, such as Google Docs. This feature not only makes excellent collaboration for common users, but can also very effective in programming.

There are also a wide variety of web-based systems that provide collaboration. For example is EtherPad that allows real-time text editing. Ace, CodeMirror are web-based text editing component which designed to be embedded into the IDE or application[3,7].

Project or software development requires the coordination and collaboration between programmers, so that the collaboration systems are very useful to improve the efficiency in making project. The effectiveness of collaboration in programming can improve the productivity and quality of project or software[8].

Collaborative programming in real-time support programmers to work on the same programming file. Real-time system will automatically combine the code typed by a programmer without manual command from the programmer (such as update, commit)[8]. Multiple programmers enable to access and edit the same source code directory, even at the same time.

In real-time sessions programmer can collaborate with other programmers by joining and leaving a session of real-time during collaborative programming. There are steps to join and leave the session using the join protocol with two-way client to receive a request from a new client who wants to join.

To accommodate the new client request, a distributed join-protocol was designed with the following message:

- JOIN: new client send a request to the session manager for request to join an existing real-time session.
- START: session manager send to all existing clients to inform the start join-protocol procedure to receives a request from a new client.
- READY: all existing clients send information to session manager about readiness for entering the state.
- FINISH: session manager send information to all existing clients about the completion of all procedures.

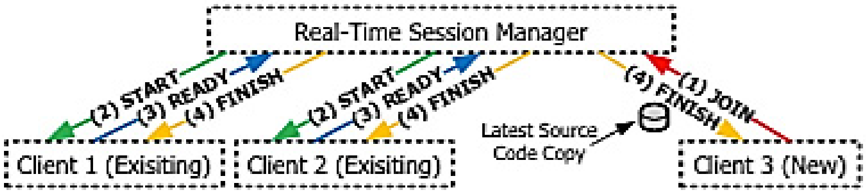


Fig. 1. Join Protocol in real-time[8]

2.2. Integrating Collaborative Program Development and Debugging within a Virtual Environment

IDEs (Integrated Development Environments) are most used tools in the activity of the programmer. To support software development and software engineering, IDE was developed in order to support collaboration in real-time, where programmers can work together to design, discuss and share in the same software development.

A collaborative integrated environment allows programmers to share programming-related tasks. One of the tools that implement collaborative IDE is ICI (Idaho Collaborative IDE)[9]. ICI allows programmers who are in different locations can collaborate in various software development activities in real-time. ICI combines technology between synchronous collaboration and a real-time debugger.

The implementation of collaborative technologies is not only used in the world of programming, but also in education. Education of computer science distance requires a more collaborative IDE integration in terms of [10]:

1) Supporting real-time collaboration in compiling, linking, running and debugging sessions
2) Providing a technology where developers can more easily communicate either by text or voice

Collaborative environment enables programmers to communicate and view the activities of other programmers in the same environment. So that programmers can chat via text or VoIP with other programmers or other project teams in real-time. To work within the same development environment, programmers can invite each other to join in a collaborative IDE session, where they collaborate together in program design, coding or debugging[9].

ICI supports collaborative software development tasks. The implementation of ICI could be used in a virtual laboratory to help teaching assistants and tutors in teaching a group of students on a remote computer. Below is the flow of information in a distributed team environment ICI that shows two-way communication such as compilers or debuggers.

2.3. Real-Time Collaborative Coding in a Web IDE

Now IDE is not only a tool that helps programmers in making projects or softwares, but the IDE now developed into tools that could help programmers communicate and collaborate with other programmers to make project more efficiently. One of the IDE that has been created to support real-time collaboration is Collabode.

Collabode is a collaborative web-based IDE that supports Java programming language. In Collabode when multiple users make changes, the result of the changes will be distributed to other users immediately without control requires programmers manually[7]. In addition, more than one user in the same project module can access this application simultaneously.

Collabode made in order to improve the quality of collaboration and project produced. To be able to collaborate on projects, user only need to visit the same URL where the other users are in. Collabode use EtherPad to support collaboration among multiple editors. To manage the projects, Collabode using Eclipse that provides syntax highlighting, compile errors and warnings, continuous compilation, code formatting and refactoring, and the execution of the program code.

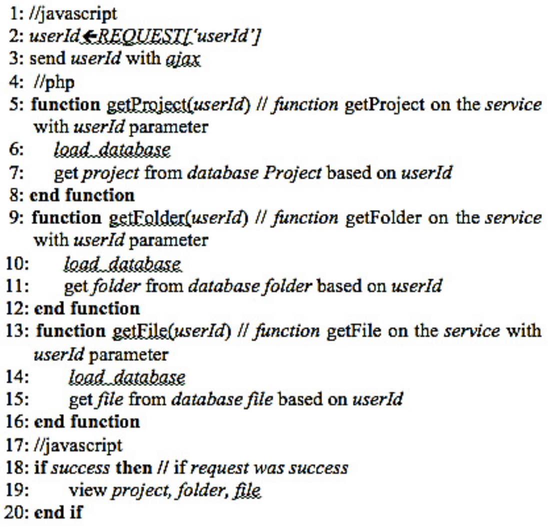
There are three interesting models used for close synchronous collaboration[7]: Test Driven Programming, Micro Outsourcing, and Mobile Instructor. The three models above give much different collaboration between one programmer to another programmer. Each of them has different ways such as using pair programming, side-by-side programming, and other collaborative models

3. Design & Method

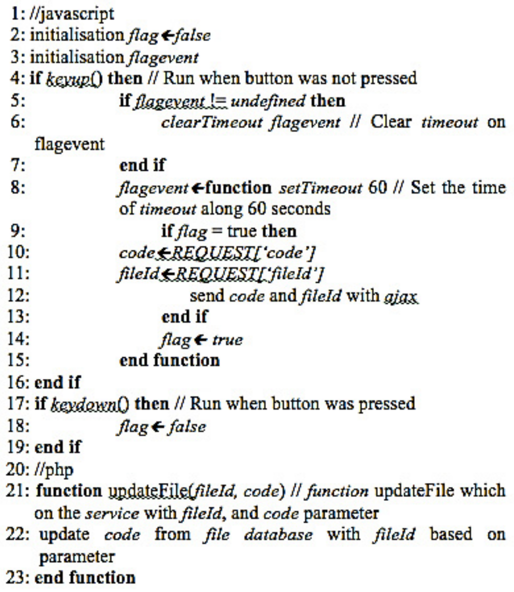
3.1. CodeR Features Algorithm

CodeR use facebook plugin for login to access the application more easily and integrate with social network to collaborate each other. CodeR supports C, C++ and Java as programming languages that users can choose as a main programming language. Users can run and build program to find out the results of the code program. Users can download the source code program along with their parent folders that has been created. Users can do collaboration to create a project together with other user in real-time. Users can communicate with other users who exist in the same project using chat as media. Users can send email to ask something or inform about bugs present in the website.

When a user successfully logs into the application, the first page displayed is the workspace page. In Figure 2 explains our algorithm that after the user logged in it will display all the projects, folders, and files that have been made (if any). Project will be displayed in the file directory on the left workspace. If a new user, then the file directory will appear empty. This application provides real-time connections so that any changes that have been made will always be updated simultaneously. Therefore this application provide auto saving feature shown in Figure 3, if users does not perform any action on the keyboard for one minute, then the code will be saved automatically.



```
//javascript
userId←REQUEST['userId']
send userId with ajax
//php
function getProject(userId) // function getProject on the service with userId parameter
    load_database
    get project from database Project based on userId
end function
function getFolder(userId) // function getFolder on the service with userId parameter
    load_database
    get folder from database folder based on userId
end function
function getFile(userId) // function getFile on the service with userId parameter
    load_database
    get file from database file based on userId
end function
//javascript
if success then // if request was success
    view project, folder, file
end if
```



```
//javascript
initialisation flag←false
initialisation flagevent
if keyup() then // Run when button was not pressed
    if flagevent != undefined then
        clearTimeout flagevent // Clear timeout on flagevent
    end if
    flagevent←function setTimeout 60 // Set the time of timeout along 60 seconds
        if flag = true then
    code←REQUEST['code']
    fileId←REQUEST['fileId']
            send code and fileId with ajax
        end if
        flag← true
    end function
end if
if keydown() then // Run when button was pressed
    flag← false
end if
//php
function updateFile(fileId, code) // function updateFile which on the service with fileId, and code parameter
update code from file database with fileId based on parameter
end function
```

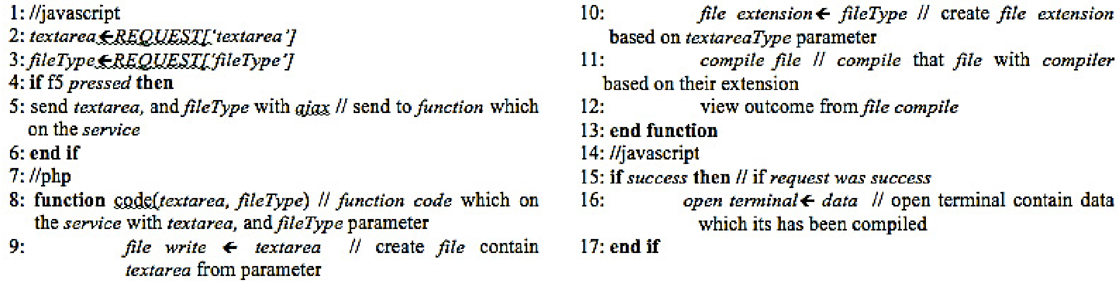
Fig. 2. Algorithm to view all project, folder, and files

Fig. 3. Algorithm to auto saving code

Run and build the program processes our algorithm shown in Figure 4. After the user create the project file and execute the program code the user will see the result. The website will display the results in the form of a terminal. If an error occurs then the terminal will display the error that occurred.

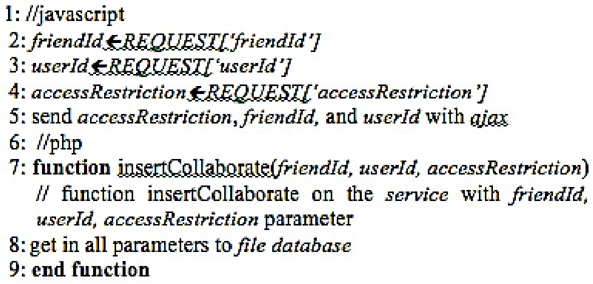
Collaboration features are very helpful in improving communication between the programmer and performance in making the project[11, 12]. Collaboration features our algorithm described in Figure 5, after makes a project file they can invite other friends. The owner of the project will send the request to another user who wants to be invited to join by giving access restrictions. So that other users can do anything according to the access granted.

Figure 6 describes the algorithms to build and display terminals. Terminals can be displayed by pressing the run button or F5 on the keyboard. When the button is pressed it will run the server NodeJS, fork terminal and the terminal will be displayed in a new tab. At the terminal can also display the error contained in the code.



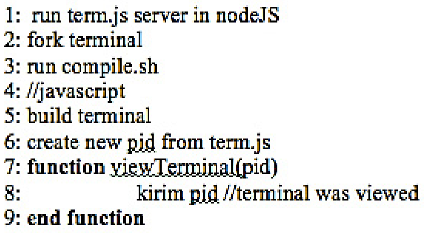
```
//javascript
textarea←REQUEST['textarea']
fileType←REQUEST['fileType']
if f5 pressed then
send textarea, and fileType with ajax // send to function which on the service
end if
//php
function code(textarea, fileType) // function code which on the service with textarea, and fileType parameter
        file write ← textarea // create file contain textarea from parameter
        file extension← fileType // create file extension based on textareaType parameter
        compile file // compile that file with compiler based on their extension
        view outcome from file compile
end function
//javascript
if success then // if request was success
    open terminal← data // open terminal contain data which its has been compiled
end if
```

Fig. 4. Algorithm to run and build program



```
//javascript
friendId ← REQUEST['friendId']
userId ← REQUEST['userId']
accessRestriction ← REQUEST['accessRestriction']
send accessRestriction, friendId, and userId with ajax
//php
function insertCollaborate(friendId, userId, accessRestriction)
  // function insertCollaborate on the service with friendId,
  userId, accessRestriction parameter
  get in all parameters to file database
end function
```

Fig. 5. Algorithm to collaboration



```
run term.js server in nodeJS
fork terminal
run compile.sh
//javascript
build terminal
create new pid from term.js
function viewTerminal(pid)
        kirim pid //terminal was viewed
end function
```

Fig. 6. Algorithm to construct terminal

3.2. Operational Transform Algorithm

The collaboration in real-time allows a group of users to access and edit the same document at the same time over a connected network. In order to connect in real-time it is needed consistency of data to be accessed at the same time, because each user operation will be executed directly without delay. To ensure consistency of copies of the shared data in collaborative real-time group-ware used operational transformation algorithm.

Operational transformation (OT) become a methods used to maintain consistency of copies of shared data on group editors[13, 14]. Operational transformation approach explains the two main components[15]: the integration algorithm and transformation functions. Integration algorithm used to receive, transmit and execute the operation. While transformation function used to combine two concurrent operations are defined on the same state.

Received operations would transform according to the local concurrent operations and then executed in operational transformation[15]. In Figure 9 illustrates the effect of the transformation function, which when op2 received on site1 then op2 must be transformed according to the op1. Position the insertion op2 is incremented because op1 has been insert f before s in the state "effect". Furthermore, op'2 executed on site1 and generates state "effects".

One of the implementation of operational transformation algorithm is the word processor application. Word processor application has a complex structure and comprehensive editing operations[14]. In addition to supporting the operation object creation and deletion, word processor application also supports updating the attributes of existing objects. As for collaborative word processor application it is important to have the ability to update collaboratively, thus speeding up the creation of documents. To support the concurrent execution of update operation is used extension operational transformation. For example the result of this method is a CoWord project that aims to convert MS Word into a real-time collaborative word processor without changing the source code.

Operational transformation has a component that is divided into two layers: the high-level transformation control algorithms, and the low-level transformation functions[14]. The strategy used in the extension of OT is to maintain a high-level control algorithm unchanged, but it adds a new update-related transformation functions. In this way makes it possible to reduce the complexity and localize the extension.

Update operation is different with Insert and Delete operation, where the update has no effect on the linear address space, while the insert and delete change the linear address space of document. When Insert transformed to concurrent Update, no change on the Insert. However, when Update is transformed to Insert, the Update's position will be increased by one if the Update's position greater than or equal to the Insert position.

4. Result

4.1. Application Architecture

User have to log in first to be able accessing the workspace. It is shown in the Figure 7. User data will be checked first whether it has been stored previously in the database. If the data exist then new log data will be saved into Collection logs, if not then new data will be created in user, file, and log Collection. Furthermore, the user will be redirected to the workspace page; project file data will be loaded from the user Collection and file database. Facebook friends list will be loaded after the user redirected into workspace. This friend's list will be used to add new collaborators, and also new collaborators who have been invited will have the ability to communicate with another users via chat.

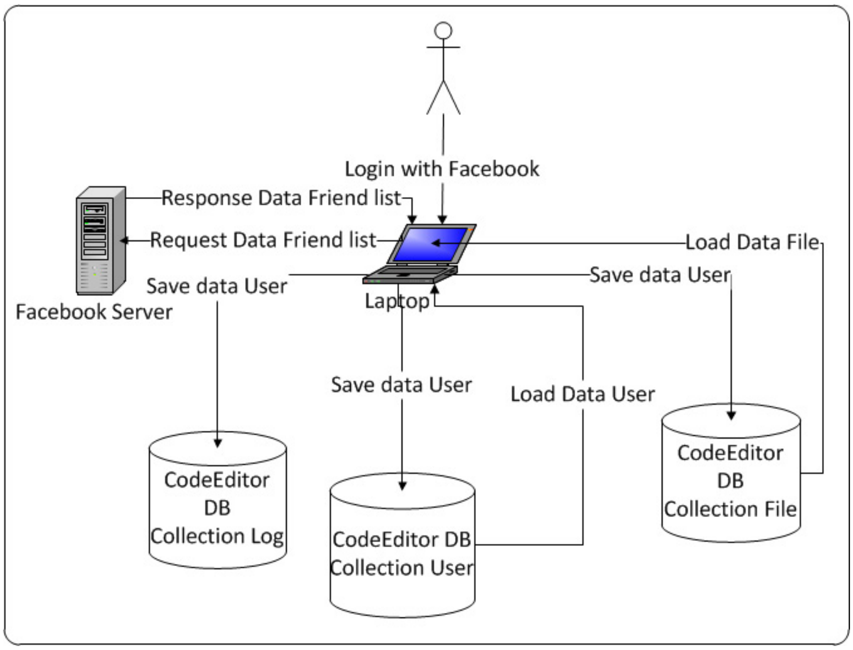


Fig. 7. Login Architecture

Figure 8 illustrates the architecture when a user sends a chat to other users and adds collaborator to join the same file project. When the user typing a message and press the enter key, the JSON data will be sent and then the chat data is saved into the chat Collection. After that the data will be sent to socket.io server and the data is returned to all users who have been invited previously. To collaborate with other users, the user could search for friends name who are registered in Facebook using Facebook auto complete engine to retrieve user data. After getting the user id, data will be inserted into file Collection and will be forwarded into socket.io server. Finally the collaborator will be put in the same file project

User can click the button placed at the left corner of the workspace page to access the folder sand the files in the workspace, and then the folders and the file will be displayed in the form of a tree. This is shown in Figure 9. Action is provided such as create, delete, rename, move the folder or file, open, download, and upload files.

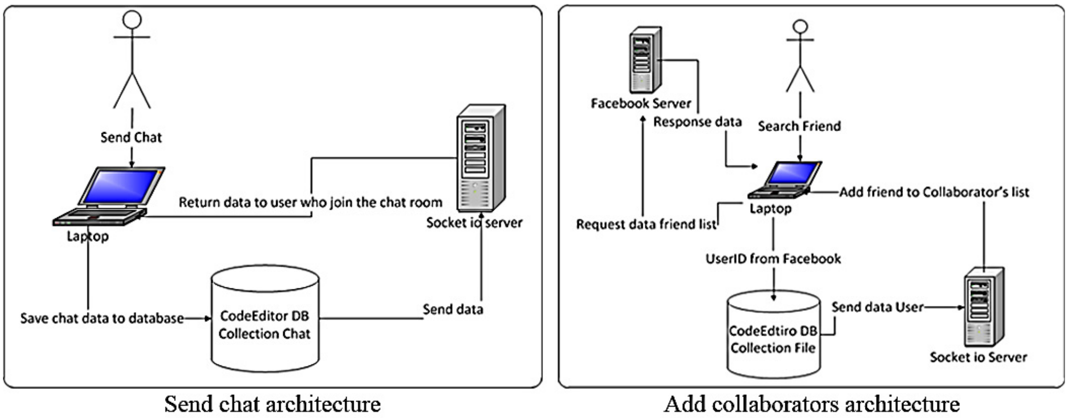

Send Chat
Laptop
Return data to user who join the chat room
Socket io server
Save chat data to database
CodeEditor DB Collection Chat
Send data
Send chat architecture
Facebook Server
Response data
Search Friend
Request data friend list
Laptop
Add friend to Collaborator's list
UserID from Facebook
CodeEdtiro DB Collection File
Send data User
Socket io Server
Add collaborators architecture

Fig. 8. Chat and add collaborator architecture

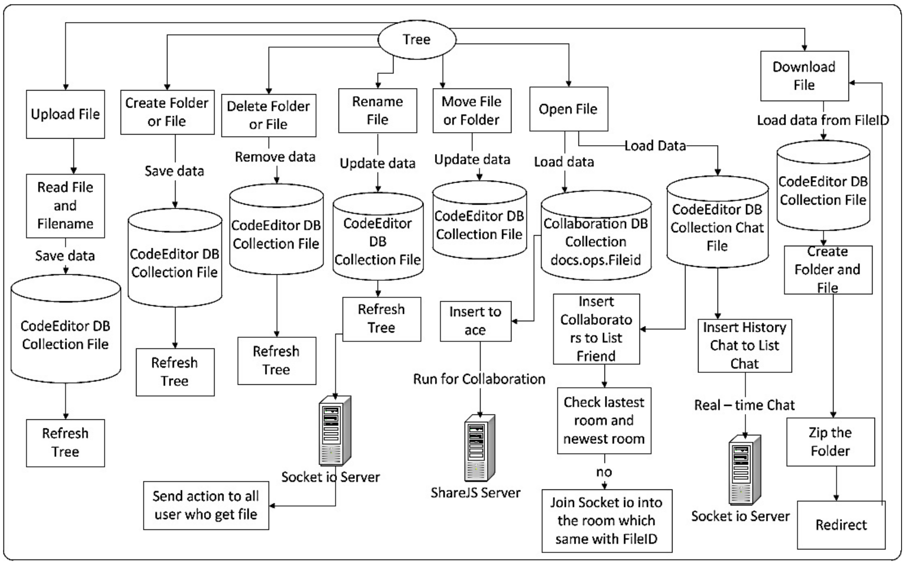

Tree
Upload File
Read File and Filename
Save data
CodeEditor DB Collection File
Refresh Tree
Create Folder or File
Save data
CodeEditor DB Collection File
Refresh Tree
Delete Folder or File
Remove data
CodeEditor DB Collection File
Refresh Tree
Rename File
Update data
CodeEditor DB Collection File
Refresh Tree
Socket io Server
Send action to all user who get file
Move File or Folder
Update data
CodeEditor DB Collection File
Insert to ace
Run for Collaboration
ShareJS Server
Open File
Load data
Load Data
Collaboration DB Collection docs.ops.Fileid
Insert Collaborators to List Friend
Check lastest room and newest room
no
Join Socket io into the room which same with FileID
CodeEditor DB Collection Chat File
Insert History Chat to List Chat
Real – time Chat
Socket io Server
Download File
Load data from FileID
CodeEditor DB Collection File
Create Folder and File
Zip the Folder
Redirect

Fig. 9. Tree project folder and file architecture

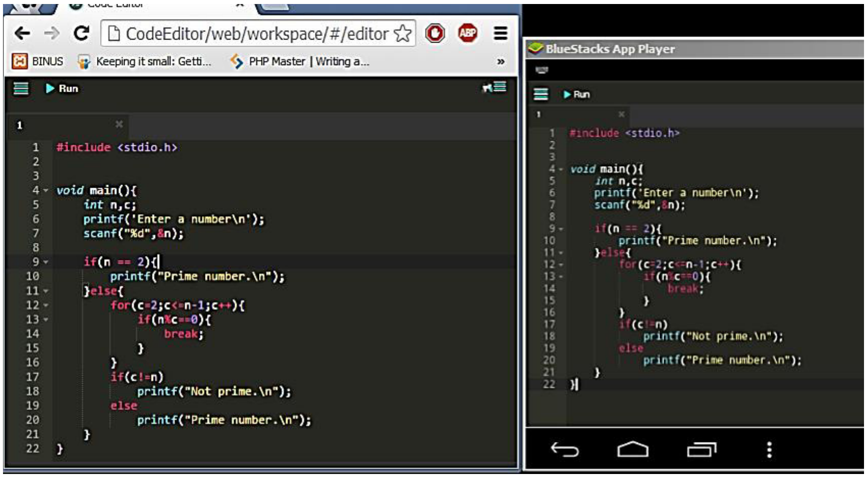
4.2. Collaboration Features

This application provides a collaboration with the user in real-time as shown in Figure 10. The user can invite other users to collaborate working on the same project through a friend list in the box on the right. After receiving an invitation to join another user in the same project, the user can communicate via chat. If a user is logged on via Facebook account, user can invite friends who are in their friend list. While login by anonymous, users will automatically connects with other users who used anonymous account. In Figure 10 shown the chat box that displays the user name, date, and there is a search feature to search friends name who would like to be invited to collaborate or communicate via chat.

5. Discussion

To evaluate the application the authors use ab (Apache HTTP Server Benchmarking Tool) or more commonly called Apache Benchmark. Apache Benchmark is a tool to measure the Apache Hypertext Transfer Protocol (HTTP) server[16]. Apache Benchmark shows how many requests per second that is capable of Apache. The type of hardware that will greatly affect the performance of the webserver is RAM.

The most basic thing, Apache Benchmark could perform load successive and/or concurrent load tests on a web page. It also could test the output such as time per request, concurrency level, transfer rate, failed request, total transferred, the time taken for the test, complete request, writing errors, HTML transferred, requests per second. Apache Benchmark provides many options such as -n (requests) for the number of requests for benchmarking sessions and -c (concurrency) for the number of multiple requests to perform at a time.


CodeEditor/web/workspace/#/editor
BINUS
Keeping it small: Getti...
PHP Master | Writing a...
Run
1
BlueStacks App Player
Run
1

```c
#include <stdio.h>


void main(){
    int n,c;
    printf('Enter a number\n');
    scanf("%d",&n);

    if(n == 2){
        printf("Prime number.\n");
    }else{
        for(c=2;c<=n-1;c++){
            if(n%c==0){
                break;
            }
        }
        if(c!=n)
            printf("Not prime.\n");
        else
            printf("Prime number.\n");
    }
}
```

```c
#include <stdio.h>


void main(){
    int n,c;
    printf('Enter a number\n');
    scanf("%d",&n);

    if(n == 2){
        printf("Prime number.\n");
    }else{
        for(c=2;c<=n-1;c++){
            if(n%c==0){
                break;
            }
        }
        if(c!=n)
            printf("Not prime.\n");
        else
            printf("Prime number.\n");
    }
}
```

Fig. 10. Collaborate between web and mobile user

This test hits the URL 1000 times (-n 1000) with maximum concurrency of 1000 simultaneous requests (-c 1000). Tests conducted using two kinds specification of computer and the following two specifications show in Table 2.

Table 2. Specification of the computer used

TYPE	PROCESSOR	RAM
SPEC 1	CPU Pentium Dual Core E2180	4 GB
SPEC 2	Core i7 2600	8 GB

Results of experiments with computer spec 1:

- Total data transferred is 3575000 bytes for 1000 requests
- Test completed in 28.842 seconds. No failed requests
- Requests per seconds: 34.67
- Time per request: 28841.650 milliseconds (for 1000 concurrent requests). So across all requests it is 28841.650 ms/100 = 28842 ms
- Transfer rate: 121.05 [Kbytes/sec] received

Results of experiments with computer spec 2:

- Total data transferred is 3501000 bytes for 1000 requests
- Test completed in 0.442 seconds. No failed requests
- Requests per seconds: 2264.06
- Time per request: 441.684 milli-seconds (for 1000 concurrent requests). So across all requests it is 441.684 ms/100 = 0.442 ms
- Transfer rate: 7740.70 [Kbytes/sec] received

Apache Bench shows the average load time (in ms) of the 1000 request. So the average load on this first test is 14.7 seconds (14 708 ms), and the longest load time is 24.9 seconds (24 988 ms). And in the second test shows the connection time was shown in Figure 19, where the mean column shows the average load time (in ms) of the 1000 request. So the average load on this first test is 0.088 seconds (88 ms), and the longest load time is 0.435 seconds (435 ms).

For further research and development the authors will suggest a few features. Create UML design automatically from the code which has been written before. Currently CodeR could only edit a single file, it would be useful if the next implementation could also edit the program code as multiple files, as if the user searches a particular keyword, the user can replace all the code which is placed in the same project in place. CodeR also could be developed into a repository-like (like GitHub) to accommodate projects that have been made as well as other projects.

6. Conclusion

This paper introduced our design and implementation of the real-time application for collaboration. CodeR is a web application that helps programmers to create and see the result of the executed source code by terminal, collaborate in real-time with other programmers by chat or invite to join the same project and manage the project such as import, export, shared projects. CodeR supports C, C ++, and Java programming languages. CodeR has the main features: provide workspace to make, execute and build the source code, real-time collaboration, chat, and build the terminal.

References

[1] M. Doernhoefer, “Surfing the Net for Software Engineering Notes", ACM SIGSOFT Software Engineering Notes, Vol. 24, No. 3, **(1999)**, pp. 15–24.

[2] L. C. L. Kats, R. G. Vogelij, K. T. Kalleberg, and E. Visser, “Software development environments on the web", in Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software - Onward! ’12, **(2012)**, pp. 99.

[3] M. Goldman, “Role-based interfaces for collaborative software development", in Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology - UIST ’11 Adjunct, **(2011)**, pp. 23.

[4] F. Frößler, "A Practice Theoretical Analysis of Real Time Collaboration Technology: Skype and Sametime in Software Development Projects", Göttingen: Cuvillier, **(2008)**.

[5] S. Klein, N. Vehring, and M. Kramer, “Introducing Real Time Communication: Frames, Modes & Rules", in Proceedings 23nd Bled eConference eTrust: Implications for the Individual, **(2010)**, pp. 591–606.

[6] K. Riemer and F. Frößler, “Introducing Real-Time Collaboration Systems: Development of a Conceptual Scheme and Research Directions", Communications of the Association for Informations Systems (CAIS), Vol. 20, **(2007)**, pp. 204–225.

[7] M. Goldman, G. Little, and R. C. Miller, “Real-time Collaborative Coding in a Web IDE", in Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, **(2011)**, pp. 155–164.

[8] H. Fan, C. Sun, and H. Shen, “ATCoPE: Any-time Collaborative Programming Environment for Seamless Integration of Real-time and Non-real-time Teamwork in Software Development", in Proceedings of the 17th ACM International Conference on Supporting Group Work, **(2012)**, pp. 107–116.

[9] H. Bani-Salameh, C. Jeffery, Z. Al-Sharif, and I. Abu Doush, “Integrating Collaborative Program Development and Debugging within a Virtual Environment", in Proceedings of the 14th Collaboration Researchers’ International Workshop on Groupware, Vol. 5411, **(2008)**, pp. 107–120.

[10] A. Sarma, “A Survey of Collaborative Tools in Software Development, Technical Report, UCI-ISR-05-3", Irvine, California, **(2005)**.

[11] S. Goel and V. Kathuria, “A Novel Approach for Collaborative Pair Programming", Journal of Information Technology Education, Vol. 9, **(2010)**, pp. 183–196.

[12] H. B. Salameh and C. Jeffery, “Collaborative and social development environments: a literature review", International Journal Computer Applications in Technology, Vol. 49, No. 2, **(2014)**, pp. 89.

[13] S. Kumawat, M. T. Scholar, and A. Khunteta, “A Survey on Operational Transformation Algorithms: Challenges, Issues and Achievements", International Journal of Engineering Science and Technology, Vol. 2, No. 7, **(2010)**, pp. 3311–3319.

[14] D. Sun, S. Xia, C. Sun, and D. Chen, “Operational Transformation for Collaborative Word Processing", in Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, **(2004)**, pp. 437–446.

[15] H. S. Molli, P. Molli, and G. Oster, “Semantic Consistency for Collaborative Systems", in Proceedings of the International Workshop on Collaborative Editing Systems - CEW 2003, **(2003)**.

[16] J. Sung-Jae, B. Yu-Mi, and S. Wooyoung, “Web Performance Analysis of Open Source Server Virtualization Techniques", International Journal of Multimedia and Ubiquitous Engineering, Vol. 6, No. 4, **(2011)**, pp. 45–52.