

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/380214482>

Design and Development of Real-time Code Editor for Collaborative Programming

Article in IARJSET · April 2024

DOI: 10.17148/IARJSET.2024.11447

CITATIONS

0

READS

869

7 authors, including:



Soumya Mazumdar

Indian Institute of Technology Madras

14 PUBLICATIONS 2 CITATIONS

SEE PROFILE



Disha Haldar

3 PUBLICATIONS 0 CITATIONS

SEE PROFILE



Shivam Chowdhury

West Bengal University of Technology

7 PUBLICATIONS 1 CITATION

SEE PROFILE



Ahana Bhattacharjee

Gargi Memorial Institute of technology

3 PUBLICATIONS 1 CITATION

SEE PROFILE

Design and Development of Real-time Code Editor for Collaborative Programming

**Soumya Mazumdar¹(0009-0006-3521-9557), Prof. (Dr.) Sayantani Das²,
Prof. (Dr.) Saurav Naskar³, Shivam Chowdhury⁴(0009-0000-1257-1757), Disha Haldar⁵,
Ahana Bhattacharjee⁶, Anjan Das⁷**

Indian Institute of Technology Madras, Tamil Nadu, India¹

Gargi Memorial Institute of Technology [JIS Group], West Bengal, India¹⁻⁷

Abstract: The Internet is expanding quickly, and many desktop apps are already being developed for the online. Using the Internet, many apps may be easily accessible at any time and from any location. Code editors are one of the tools that developers require to design their applications. The goal of this research is to employ web socket technology to build and develop a real-time code editor application that facilitates user collaboration while working on the project. Users of this program can work together in real time on a project thanks to its capability. The authors employed an analysis process that includes studying the literature, and studying the code editor software that are currently in use. CrossCode is a web application that offers a workspace for creating, performing, and collaborating with other users in real-time over the terminal. The primary functionalities of the application include a workspace for creating, running, and building source code as well as real-time chat and terminal building. The programming languages C, C++, Python, and Java are supported by this application. The current research contributes to the advancement of online code collaboration tools, providing developers with an efficient and accessible platform for remote teamwork and code sharing.

Keywords: Real-time Code Editor, Web Socket Technology, Collaborative Coding, Multi-language Support, Cloud-based Code Development

I. INTRODUCTION

As software design began to move from desktop to online platforms, software engineering saw a trend of technological improvement. The need to install plug-ins and the lengthy configuration process before a software can be used are only two of the many disadvantages of desktop IDEs, despite the abundance of IDE (Integrated Development Environment) tools available today, such as Eclipse and Visual Studio [1]. This problem may be a major time waster when there are several devices that need to be configured [2].

An increasing number of software applications run in the cloud and utilize a web browser as their user interface. This removes the requirement for desktop computer installation and setup and allows for immediate collaboration across users [2].

Using a single integrated development environment (IDE) is one method for real-time teamwork, much like pair programming. Pair programming is the practice of allowing two programmers to access and collaborate on the same code inside a single working environment [3]. When working in pairs, programmers may create, modify, and delete source code in real time. Pair programming might help with program code synchronization difficulties by enabling any programmer working on the same project to know who made changes to the code and track when they occur.

Collaborative technologies might be useful for programmers who work remotely but in the same environment to discuss or correct bugs in the application. It must thus create an application that improves the speed of program development, such real-time collaboration, and utilize a terminal to construct, execute, and show the result of the program.

1.1 Real-time Communication

Real-time communication (RTC) is facilitated by a range of collaborative applications that are integrated with communication technologies such as voice-over-IP (VoIP) telephony and instant messaging [4]. RTC technologies are composed of four interrelated building blocks: contextualization, presence awareness, unified communications, and the E-Collaboration portfolio [5]. RTC systems often make two-person and multi-person conferences possible.

By combining many synchronous communication channels into one integrated environment, RTC solutions allow users to collaborate in real time [6]. This covers phone conversations, document editing, and multi-person video conferencing. RTC technology may help with the synchronization problem, especially when used in group environments. Table 1 lists the components of the RTC systems. RTC systems provide users autonomy and the capacity to interact with others at a distance. Additionally, they have enhanced and promoted communication and engagement, which may aid in the project's more effective growth.

Table 1. Building Blocks of Real-Time Collaboration Systems

Unified Communication (UC)	Presence Information Portfolio	Contextualization	E-Collaboration
<ul style="list-style-type: none"> • Various media and communication channels • Media and device integration • Rule-based configuration of message routing and call diversion • Definition preferred media • Unified messaging portal 	<ul style="list-style-type: none"> • Signaling of people and communication channel • Aggregation of presence information in group, role, and objective level • Active management of buddy list. • Individualize and automatic signaling 	<ul style="list-style-type: none"> • Integration with applications • Context specific buddy lists • Mobile RTC with location based services 	<ul style="list-style-type: none"> • Audio, video and conferences seminars • Ad Hoc application sharing • Team calendars and contact management • Joint whiteboards and discussion forums • Document folders

1.2 State of the Art

The objective of this project is to create code editor programs that can be used for chat as a means of user communication, syntax checking, compiling and executing source code via a terminal, and collaborative work on a specific file. This program provides a workspace with useful features including autocomplete for C, C++, Python, and Java in addition to the ability to create, modify, execute, and compile pre-written source code. Users who want to export source code files to their desktop or import project files from the desktop may utilize the upload and download functionalities. This tool has the potential to enhance project development efficiency and facilitate real-time collaboration among programmers on projects.

The remainder of the document is structured as follows: A thorough overview of the literature is given in Section 2, which also covers the development of collaborative tools historically and looks at important features, communication protocols, and difficulties with real-time code editors. The research methods and design concepts used in this work are described in depth in Section 3. After that, Section 4 explores the implementation stage and offers insights into the difficulties and technical concerns encountered. After the results are presented in Section 5, a thorough discussion that links our findings to previous research and considers ramifications follows. In Section 6, the study wraps up by providing a succinct overview of the major contributions and outlining potential directions for further research into collaborative programming tools.

II. RELATED WORK

The need for effective cooperation in software development has led to a fast evolution in the collaborative programming environment. This study of the literature examines recent findings and advancements in the creation of real-time code editors, with an emphasis on how these tools support group programming projects. A software platform or tool created to support collaborative learning experiences focused on solving programming problems is known as problem-based collaborative learning groupware, and it may be used to improve computer programming abilities. [7] The design of a contemporary online interview platform for a recruiting system, as described by Author [8], entails combining a number of elements to speed up the hiring process, enhance the applicant experience, and provide recruiters insightful information. Whether working in a big, remote development environment or as part of a small team, collaboration tools are critical for developers working on projects [9]. Through the development of a supportive learning environment, the improvement of critical thinking abilities, and the provision of a more comprehensive grasp of OOP principles, collaborative programming may have a favorable influence on students' progress in learning object-oriented programming [10].

An IDE add-on called Improv [11] is used to create and deliver code-based presentations. In order to evaluate instructors' preparedness for blended learning, authors [12] described the difficulties they encounter while teaching online. The research undertaken by the authors [13] aims to investigate and determine the degree of Information and Communication Technology (ICT) competency among undergraduate students at public institutions located in Nigeria.

2.1 Real-time Collaborative Programming

Many programmers collaborate in teams while working on development projects. Any coder having access to the code may add, modify, and write new code all inside the same project file. Hence, in order to avoid code duplication, programmers must synchronize, and solving this synchronization issue requires integrated real-time collaboration in a single environment.

The Integrated Development Environment (IDE) is designed to provide programming teams a real-time collaborative environment in which they may edit text, execute and build code, collaborate, and do a lot of other things. Because text may be edited in real-time, many users can work together to modify a document and instantaneously share the changes with other users who have access to the same document.

There are many free tools that allow you to modify text in real time, such as Google Docs. Not only does this capability facilitate excellent communication for regular users, but it may also be quite helpful in programming.

Furthermore, a wide range of web-based solutions are accessible for teamwork. similar to EtherPad, which allows for real-time text editing. IDEs and other apps are designed to be coupled with web-based text editing tools such as Ace and CodeMirror.

Collaboration technologies play a critical role in optimizing project creation efficiency, since software development projects need coordination and collaboration between programmers (14). The productivity of a project or software segment may be increased by the effectiveness of cooperation in programming.

Programmers may collaborate in real time on a single programming file by using real-time collaborative programming. Without the programmer's human involvement, the real-time system [15] will automatically integrate code that is typed by the programmer. Multiple programmers may access and modify the same source code directory at the same time.

Programmers may collaborate by joining and leaving real-time sessions while working on collaborative programming. There are steps to take in order to utilize the connection protocol with two-way clients to enter and quit a session when a new client wishes to join.

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

- **JOIN:** By submitting a request to the session manager, a new client may ask to join an ongoing real-time session.
- **START:** The join-protocol process is started when the session management alerts each of the existing clients that a new client request has been received.
- **READY:** All present clients provide information to the session manager on their preparedness to enter the state.
- **FINISH:** The session management informs each and every one of the presently enrolled customers that all processes have been finished.

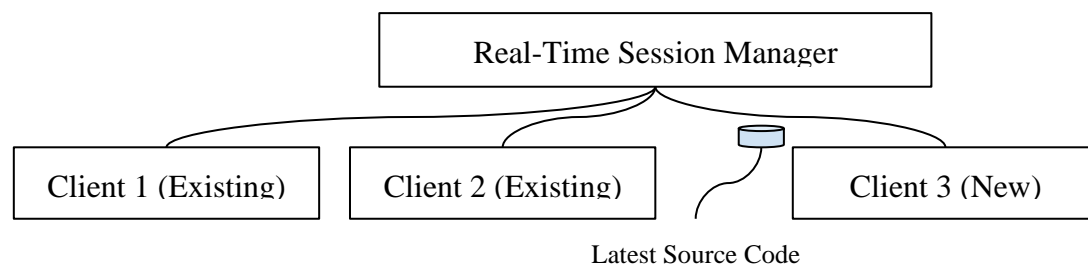


Fig. 1. Real time Protocol [15]

2.2 Integrating Collaborative Program Development and Debugging in a Virtual Environment

IDES (Integrated Development Environments) are the most often used tool among programmers. Programmers may work together in real-time on a single project, debating, planning, and sharing it thanks to the integrated development environment (IDE), which was created to support real-time collaboration in software development and engineering.

Programmers may exchange programming-related tasks in an integrated collaborative environment. One technology used to implement collaborative IDE is called ICI (Idaho Collaborative IDE) [16]. Because ICL ICT combines synchronous collaboration technology with a real-time debugger, programmers may collaborate in real-time on a range of software development activities even if they are situated in different places.

The domains of programming and education are implementing collaborative technology. Remote computer science education demands a more collaborative IDE integration to enable real-time participation throughout the compilation, linking, running, and debugging of sessions. This connection may provide a platform that facilitates textual or voice conversation amongst developers.

When working in a collaborative environment, programmers may communicate with one another and keep an eye on what other programmers are doing in the same area. Programmers may use text or voice chat to interact in real time with other programmers or project teams. In order to collaborate on program design, coding, or debugging inside the same development environment, programmers may invite one another to take part in collaborative IDE sessions [16].

2.3 Real-Time Collaborative Coding in a Web IDE

IDEs have changed dramatically over time, moving from standalone coding tools to teamwork environments. Their responsibilities have grown to include helping developers of specific projects as well as promoting smooth communication and cooperation between programmers. Collabode, an IDE made especially for real-time collaboration, is one such example. The trend toward collaborative integrated development environments (IDEs) is a reflection of the increasing importance of cooperation in maximizing the effectiveness of the software development process.

Online collaboration is made easier using Collabode, an integrated development environment (IDE) for Java. With Collabode, adjustments made by many users may be quickly shared with other users without requiring manual programming [14]. Furthermore, this application is available to numerous users concurrently inside a single project module.

The goal of cooperation is to improve both teamwork and the quality of the project result. To cooperate on projects, users just need to visit the same URL as one another. Collabode uses EtherPad to make editor collaboration easier. Collabode offers capabilities including syntax highlighting, continuous compilation, compile errors and warnings, code formatting and refactoring, and program code execution. Collabode utilizes Eclipse to manage the projects.

Three methods are used for limited synchronous collaboration: Test Driven Programming, Micro Outsourcing, and Mobile Instructor [17]. The three models mentioned above illustrate the many ways in which programmers might work together. They all use various strategies, such as pair programming, side-by-side programming, and other cooperative techniques.

III. METHODOLOGY AND DESIGN

The study's design and methods are described in this section. The CrossCode Features Algorithm is examined in Section 3.1, while the Operational Transform Algorithm is examined in Section 3.2. These algorithmic frameworks play a crucial role in the study's overall design and functioning, laying the groundwork for further research and analysis.

3.1 CrossCode Features Algorithm

In order to streamline user experience, CrossCode employs the ReactJS plugin for login and interfaces with social networks to enable collaboration. In CrossCode, users have the option to choose Java, C, C++, Python, and C as their main programming languages. Users may run and develop the code program to observe the outcomes. Users may obtain the source code program and the created parent folders. By collaborating, users may work on projects in real time. Chat provides a communication channel for users involved in the same project. If users have any queries or concerns, they may contact the website.

As soon as a user connects into the software successfully, the workspace screen displays. After the user signs in, all finished projects, folders, and files—if any—will be shown according to the method depicted in Figure 2.

```

1 //javascript
2 user REQUEST['userId']
3 send userId with ajax
4 //php
5 function getProject(userId) // function getProject on the service with
   userId parameter
6 load database
7 get project from database Project based on userId
8 end function
9 function griEulden(userId) // function getFolder on the service with
   userId parameter
10 load database
11 get folder from database folder based on userId
12 end function
13 function getFile(surid) // function getFile on the service with
   userId parameter load database
14 get file from database file based on userId
15 end function
16 //javascript
17 if success then // if request was success
18 view project, folder, file
19 end if

```

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

The project is visible in the file directory of the workspace on the left. If this is a new user, the file directory will seem empty. This application offers real-time connectivity, so all changes are updated at the same time. This program, therefore, includes the auto-saving function shown in Figure 3, which will automatically save the code in the event that the user doesn't press the keyboard for a minute. Figure 4 shows the method to execute and construct the application. After generating the project file and starting the program, the user will be able to see the results. The results will be shown as a terminal on the website. If an issue occurs, the error message will show up on the terminal.

```

1 //javascript
2 initialisation flag <- false
3 initialisation flagevent
4 If keyup() then // Run when button was not pressed
5   if flagevent != undefined then
6     clearTimeout flagevent // Clear timeout on flagevent
7   end if
8   flagevent <- function setTimeout 60 // Set the time of rimeout
   along 60 seconds
9   if flag true then
10    code <- REQUEST ['code']
11    field <- REQUEST ['field']
12    send code and fileld with ajax
13  end if
14  flag <- true
15  end function
16 end if
17 If keydown() then // Run when button was pressed
18   flag <- false
19 end if
20 //php
21 function updateFile(fileld, code) // function updateFile which on the
   service with fileld, and coole parameter
22 update code from file database with fileld based on parameter
23 end function

```

Figure 3. Algorithm to auto saving code

```

1 //javascript
2 textarea <- REQUEST ['textarea']
3 file Type <- REQUEST['fileType']
4 if f5 pressed then
5 send textarea, and fileType with ajax //send to faction which on the
  service
6 end if
7 //php
8 function code (textarea, fileType) //function code which on the arvion
  with textarea, and fileType parameter
9 file write <- text area //create file contain tint area from paramete
10 file extension <- filetype //create file extension based on textarea Type
  parameter
11 compile file //compile that file with compler based on their extension
12 view outcome from file compile
13 end function
14 //javascript
15 if success then //if request was successful
16 open termittal data //open terminal contain data which it has been
  compiled
17 end if

```

Figure 4 Algorithm to run and build program

Collaboration features are very helpful in improving communication between the programmer and performance while developing a project [19]. Figure 5 illustrates the collaborative aspects of our system, which allow friends to be invited after they create a project file. In order for other users to behave in line with the access granted, the project owner will enforce access limitations by forwarding the request to another user who wants to get an invitation to join. The terminal creation and display algorithm is shown. as seen in Figure 6. By pressing the run button or the F5 key on the keyboard, users may see terminals. Pressing the button will cause the terminal to open in a new tab and run the NodeJS server. The terminal is the code error that is present.

```

1 run term.js server in nodeJS
2 fork terminal
3 run compile.sh
4 //javascript
5 build terminal
6 create new pid from term.js
7 function view Terminal(pid)
8     kirim pid //terminal was viewed
9 end function

```

Figure 5. Algorithm to collaborate among different users.

```

1 //javascript
2 friendId <- REQUEST['friendid']
3 userId <- REQUESTI ['userId']
4 accessRestriction <- REQUEST['accessRestriction']
5 send accessRestriction, friendId, and userId with ajax
6 //php
7 function insertCollaborate(friendId, userId, accessRestriction) //
  function insertCollaborate on the service with friendId, userId,
  accessRestriction parameter
8 get in all parameters to file database
9 end function

```

Figure 6. Algorithm to build and display terminal

3.2 Operational Transform Algorithm

Through real-time collaboration, many people may see and edit the same document at the same time over a network connection. In order to connect in real-time, consistency of data must be collected concurrently since every user action would be executed instantly. In collaborative real-time groupware, operational transformation methods are used to ensure consistency of copies of shared data. Operational transformation (OT) is one method for guaranteeing consistency among copies of shared data on group editors [20]. The operational transformation approach clarifies the two main components, which are the integration algorithm and transformation functions [21]. An integration algorithm executes, transmits, and receives the process. In contrast, the transformation function unifies two running processes by defining them in the same state.

Receiving operations would then execute operational transformation in line with local contemporaneous operations [21]. One application of the operational transformation approach is the word processor software. A word processor program has a complex structure and can carry out a wide range of editing tasks [22]. Word processing software facilitates not only the addition and removal of objects but also the alteration of an object's properties. For collaborative word processing apps, the ability to update collaboratively is essential since it speeds up document creation. The concurrent execution of the update operation is supported by the use of extension operational transformation. For example, this method produced the CoWord project, which aims to convert Microsoft Word into a real-time collaborative word processor without changing the source code. The low-level transformation functions and the high-level transformation control algorithms are the two levels that comprise the operational transformation component [22]. The OT expansion uses a technique that adds a new update-related transformation function without changing the high-level control algorithm. This strategy makes it possible to reduce complexity and localize the extension.

In contrast to the insert and delete operations, the update operation modifies the document's linear address space, while the latter does not. Insert switched to a concurrent update, yet it stayed the same. However, if the Update's position is more than or equal to the Insert position, its position will be raised by one when it is turned to Insert.

IV. IMPLEMENTATION

The use of CrossCode is covered in this section. The user has to log in before they may access the workspace. First, to determine if the user's data has ever been stored in the database, it will be reviewed. If the log data already exists, it will be recorded in the collection logs; if not, it will be produced in the user, file, and log collections. Moreover, the user will be sent to the workspace page, where project file data from the user's collection and file database will be imported. The user's Facebook friends list loads when they are routed to the workplace. New collaborators will be invited via the friend's list, and those who accept will be able to communicate with each other. When a user starts a conversation with another user, Figure 8 displays the architecture. When a user inputs a message and presses the enter key, JSON data is transmitted, and the chat data is then stored in the chat collection. After that, the data is sent to the socket.io server and then made available to all of the users that were first invited. Figure 9 illustrates how to add a contributor to a project using the same file. The user may search for the names of friends who have registered for CrossCode in order to collaborate with other users. After obtaining the user ID, data will be placed into the file collection and transferred to the socket.io server. Finally, the collaborator will be appended to the same file for the project. Users may click the button in the left corner of the page to access the folders and files in the workspace. A tree-like representation of the files and folders will materialize.

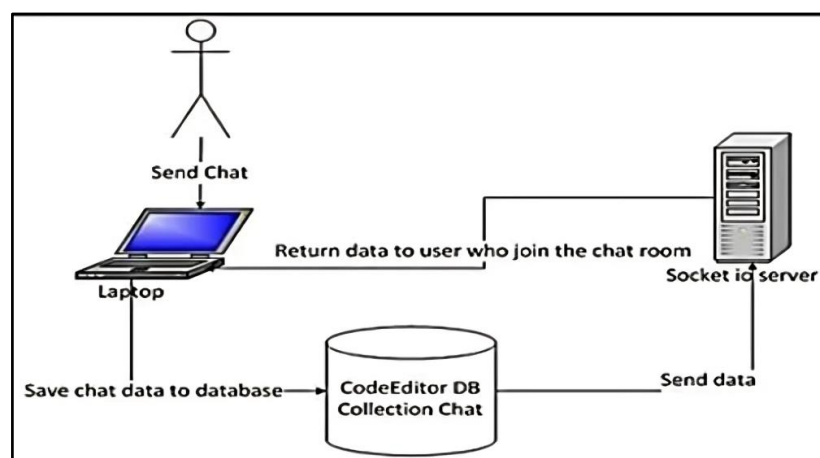


Figure 8. Chat architecture

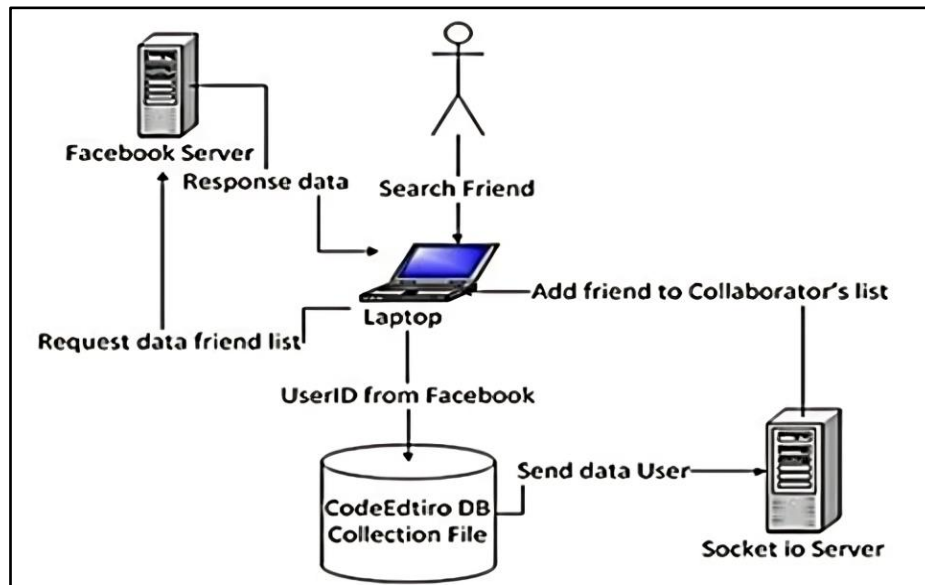


Figure 9. Add collaborator architecture

Figure 10 illustrates how real-time user engagement is possible with this application. Real-time user engagement is possible with this application. The user may invite other users to collaborate on the same project by using the friend list shown in the box on the right. After receiving an invitation to collaborate on the same project, the user may communicate with other users using chat. If someone is logged in using their Facebook account, they may send invitations to friends who are on their friend list. When users log in, other users who have used anonymous accounts will immediately connect with them. Figure 10 displays a chat window with the user's name, the current date, and a search function for finding friends who are willing to be contacted via chat to work together or talk.

```

1 #include <stdio.h>
2 int main()
3 {
4     int n, i, c=0;
5     printf("Enter a number\n");
6     scanf("%d",&n);
7     for(i=1;i<=n;i++)
8     {
9         if(n%i==0)
10            c++;
11     }
12     if(c==2)
13         printf("Prime number");
14     else
15         printf("Not Prime number");
16     return 0;
17 }

```

Figure 10. Collaboration between web and mobile user

V. RESULTS AND DISCUSSION

This study uses the Apache Benchmark, sometimes referred to as the ab (Apache HTTP Server Benchmarking Tool), which is a reliable instrument for evaluating our application's performance. Apache Benchmark is renowned for its ability to accurately gauge the Apache Hypertext Transfer Protocol (HTTP) server's performance. In this work, we use ab to do a comprehensive analysis of our application, including its overall effectiveness, scalability, and responsiveness. The Apache Benchmark shows the maximum number of requests per second that Apache can process. The hardware type that will most significantly impact the web server's performance is RAM.

First and foremost, Apache Benchmark could load test a website both sequentially and concurrently. The time spent on each request, the level of concurrency, the transfer rate, the number of failed requests, the total number of transfers, the length of the test, the completed requests, the mistakes in writing, the HTML, the transfers, and the requests per second could all be tested. Apache Benchmark has several parameters, such as -e for the number of requests to be handled simultaneously and n for the number of requests for benchmarking sessions.

1000 requests must be sent to the designated URL during this assessment, with a maximum of 1000 requests being sent at once. Two different computer setups, each with its own set of requirements, are used to conduct the testing. The machine has a Pentium Dual Core E2180 CPU and 4 GB of RAM for SPEC 1. SPEC 2, on the other hand, has an 8 GB RAM configuration and a Core i7 2600 CPU. Table 2 displays the experimental outcomes.

The 1000 requests' average load time (in milliseconds) is shown by Apache Bench. According to the ApacheBench findings for the first test, the greatest load time was 23.8 seconds (23.388 ms), while the average load time for 1000 requests was 14.6 seconds (14,608 ms). The greatest load time in the second test, which measures connection time, is 0.335 seconds (335 ms), but the average load time is much reduced at 0.078 seconds (78 ms). These outcomes provide light on how well the application functions in various testing scenarios.

Comprehensive examination of the Apache Benchmark results provides valuable insights into the overall effectiveness, scalability, and responsiveness of the assessed application. These results provide a more thorough insight of the elements impacting the application's performance environment, in addition to the apparent hardware-dependent performance variations between SPEC 1 and SPEC 2. These conclusions might be the starting point for more research and development that could optimize the program, enhance user satisfaction, and promote more economical use of resources. This iterative approach highlights how important it is to keep assessing and adjusting the application to make sure it meets evolving user requirements and performance targets.

VI. CONCLUSION AND FUTURE SCOPE

This article presents the real-time collaboration application's concept and implementation. CrossCode is an online tool that offers project management features including import, export, and shared projects in addition to enabling real-time chat conversation amongst programmers and inviting them to join a project. Programmers may also use it to create code and see the output of their work when it runs. CrossCode supports the programming languages C, C++, Python, and Java. CrossCode's main features include a workspace for writing, testing, and developing source code in addition to chat, terminal building, and real-time collaboration.

The capacity to automatically produce UML designs from written code is one of the characteristics that the authors will suggest further research and development for. CrossCode can only modify a single file at this time. In a future release, it would be useful to have many files where the program code may be modified. In this manner, if the user looked up a certain phrase, they could change all the code in the same project at once. CrossCode may eventually be evolved into a system that functions as a repository, akin to GitHub, for managing both completed and in-progress projects.

REFERENCES

- [1] Doernhoefer, M. 2016. Surfing the net for software engineering notes. *Software Engineering Notes*. 41, 5 (Nov. 2016), 11–18. DOI:<https://doi.org/10.1145/2994205.2994209>.
- [2] Kats, L.C.L., Vogelij, R.G., Kalleberg, K.T. and Visser, E. 2012. Software development environments on the web. *Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. (Oct. 2012). DOI:<https://doi.org/10.1145/2384592.2384603>.
- [3] Goldman, M. 2011. Role-based interfaces for collaborative software development. *Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology*. (Oct. 2011). DOI:<https://doi.org/10.1145/2046396.2046410>.
- [4] Fröbler, F. 2008. *A practice theoretical analysis of real time collaboration technology: Skype and Sametime in software development projects*. Cuvillier Verlag.
- [5] Klein, S., Vehring, N. and Kramer, M. 2010. Introducing real time communication: frames, modes & rules. *23rd Bled eConference eTrust*. (2010), 591–606.
- [6] Riemer, K. and Fröbler, F. 2007. Introducing Real-Time Collaboration Systems: development of a conceptual scheme and research directions. *Communications of the Association for Information Systems*. 20, (Jan. 2007). DOI:<https://doi.org/10.17705/1cais.02017>.

- [7] Chorfi, A., Hedjazi, D., Aouag, S. and Boubiche, D. 2020. Problem-based collaborative learning groupware to improve computer programming skills. *Behaviour & Information Technology*. 41, 1 (Jul. 2020), 139–158. DOI:<https://doi.org/10.1080/0144929x.2020.1795263>.
- [8] Kathiravan, M., Madhurani, M., Kalyan, S., Raj, R. and Jayan, S. 2023. A modern online interview platform for recruitment system. *Materials Today: Proceedings*. 80, (Jan. 2023), 3022–3027. DOI:<https://doi.org/10.1016/j.matpr.2021.06.459>.
- [9] Jackson, V., Van Der Hoek, A., Prikładnicki, R., Ebert, C. and Ebert, C. 2022. Collaboration tools for developers. *IEEE Software*. 39, 2 (Mar. 2022), 7–15. DOI:<https://doi.org/10.1109/ms.2021.3132137>.
- [10] Krismadinata, K., Efan, Boudia, C., Jama, J. and Saputra, A.Y. 2023. Effect of Collaborative programming on Students Achievement Learning Object-Oriented Programming course. *International Journal of Information and Education Technology*. 13, 5 (Jan. 2023), 792–800. DOI:<https://doi.org/10.18178/ijiet.2023.13.5.1869>.
- [11] Chen, C.H. and Guo, P.J. 2019. Improv. *Proceedings of the Sixth (2019) ACM Conference on Learning*. (Jun. 2019). DOI:<https://doi.org/10.1145/3330430.3333627>.
- [12] Gumawe, C.M., Santiago, J.C. and Saguil, J.L. 2022. Full paper ViewGo back Challenges faced by teachers during online class: Basis for readiness in blended learning. *International Journal of Scientific Research in Multidisciplinary Studies*. 8, 8 (2022), 18–25.
- [13] Chigbundu, M.C. and Oluwabiyyt, M.O. 2023. Digital Literacy, Perception and Challenges of elearning among undergraduates in public Universities of Nigeria. *International Journal of Scientific Research in Multidisciplinary Studies*. 9, 11 (2023).
- [14] Goldman, M., Little, G. and Miller, R.C. 2011. Real-time collaborative coding in a web IDE. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. (Oct. 2011), 155–164. DOI:<https://doi.org/10.1145/2047196.2047215>.
- [15] Fan, H., Sun, C. and Shen, H. 2012. ATCOPE: Any-time collaborative programming environment for seamless integration of real-time and non-real-time teamwork in software development. *Proceedings of the 17th ACM International Conference on Supporting Group Work*. (Oct. 2012), 107–116. DOI:<https://doi.org/10.1145/2389176.2389194>.
- [16] Bani-Salameh, H., Jeffery, C., Al-Sharif, Z.A. and Doush, I.A. 2008. Integrating Collaborative Program Development and Debugging within a Virtual Environment. *Lecture notes in computer science*. 107–120.
- [17] Sarma, A., Institute for Software Research, Donald Bren School of Information and Computer Sciences, and University of California, Irvine 2005. A survey of collaborative tools in software development. *ISR Technical Report*. (Mar. 2005).
- [18] Goel, S. and Kathuria, V. 2010. A novel approach for collaborative pair programming. *Journal of Information Technology Education*. 9, (2010).
- [19] Bani-Salameh, H. and Jeffery, C. 2014. Collaborative and social development environments: a literature review. *International Journal of Computer Applications Technology/International Journal of Computer Applications in Technology*. 49, 2 (Jan. 2014), 89. DOI:<https://doi.org/10.1504/ijcat.2014.060520>.
- [20] Kumawat, S. and Khunteta, A. 2010. String based new operations –Find and replace by new operational transformation algorithms for Wide-Area collaborative applications. *International Journal of Computer Applications*. 6, 7 (Sep. 2010), 25–30. DOI:<https://doi.org/10.5120/1088-1420>.
- [21] Sun, D., Xia, S., Sun, C. and Chen, D. 2004. Operational transformation for collaborative word processing. *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. (Nov. 2004), 437–446. DOI:<https://doi.org/10.1145/1031607.1031681>.
- [22] Skaf-Molli, H., Molli, P. and Oster, G. 2016. Semantic consistency for collaborative systems. *Proceedings of the International Workshop on Collaborative Editing Systems-CEW*. (Jun. 2016).
- [23] Jung, S.-J., J., Bae, Y.-M., Soh, W., and Dept. of Computer Engineering, Hannam University, Korea 2011. *Web performance analysis of open source server virtualization techniques*.