Web-based integrated development environment for Java, Android and a new programming language

PROJECT PLAN

Supervisor
Dr. T.W. Chim

Team members
Poon Stirling Yeu: 3035069263
Shum Chi Chung: 3035070236
Wong Man Chun: 3035071955
Table Of Contents

Introduction
Objectives and Benefits
Scope
Deliverables
Approach & Methodology
Risks, Challenges & Mitigation
Division of Labour
Schedule of Tasks
Conclusion
References
1. Introduction

Productivity is one of the major issues programmers have to deal with during their software development in order to finish more software development projects in a limited timeframe. With a desire for promoting software quality, a growing need for software development and a deficiency in programmer talent, a software productivity study in 1980 suggested that an easily accessible software platform can promote software development efficiency significantly [1]. This has resulted in a demand for a multi-language cross-platform software development tool [1].

An integrated development environment (IDE) is an application that contains text editor, debugger and compiler for software development [2], [3]. Instead of writing code in a text editor such as Notepad, programmers can now code in a more user-friendly interface, with command-line console and debugging tools all in one place. Traditional IDEs including Eclipse and IntelliJ IDEA can automate syntax highlighting, code completion, error checking and code refactoring [4], [5]. Although these tools have become essential tools for developers, installation and configuration of these is often time-consuming, a common headache to programmers [6]. These tools also consume vast amounts of CPU and memory space, which may be taxing on old computers [6]. Furthermore, with the increasing trend of project team development, a majority of these single-environment development platforms fail to serve team collaboration, which makes version control and build management more difficult [6][7]. As such, the traditional form of IDE is confining the potential for a better and more efficient software development platform.

According to Software Development Environments on the Web: A Research Agenda, an online IDE, being a centralized platform for program development, serves as a solution to the above issues [7]. It avoids the installation of bulky software on local machines and eliminates installation and configuration time [8]. Also, every team player can get access to the single copy of source code on server which can avoid version conflict and prevent duplicates. This type of Software As a Service (SaaS) platform took over the software control of the development toolkit. Programmers can code on the SaaS platform any time on any computer with internet access, without worrying about storage or server [7], [8]. Lennart Kats, Richard Vogelij, Karl Trygve
Kalleberg and Eelco Visser made three points in their research paper describing their perception of an ideal online IDE [7].

Firstly, a modern code editor should have a scanner and parser that processes the source code for presentation, semantic and editing purposes [7]. The research paper referred to the article written by Lennart Kats and Eelco Visser which visualizes the typical composition of IDE parser [9]. In the presentation module, outline view can preview the structure of the source program, syntax highlighting can make syntax and operation stand out of text and code folding can expand and collapse text within the brackets or the scope of document. In the editing module, bracket matching and automatic indentation can preserve the integrity and consistency of source code structure. Syntax completion also made coding faster and more efficient. In the semantic module, name analysis takes place to resolve references and to highlight words with no occurrences. An error or warning description should be displayed if the predictate of some error validation rules, usually implemented by regular expression, is satisfied.

The structure of code parser in a typical IDE (Extracted from [9])

Secondly, an online IDE should facilitate team collaboration processes. Inside an organization, multiple developers can edit the source code at the same time. Source code can be synchronized by web browser uploading the source code changes to server which then broadcasts the changes to other online clients. Operational transformation is implemented to ensure that the version of source code stays the same on all client browsers without being affected by connection delay or sudden disconnection and re-connection of computers [10]. Issue tracking is also an essential feature for project management and progress tracking. Other related features include version control and build management system such as Git, which can ensure data consistency, facilitate branch development and allow the
restoration of previous version of source code. Group chat can also be implemented to foster group communication.

Thirdly, it is possible to implement e-learning features in the online IDE. Utilization of a cloud development platform can eliminate the difficulty of installing softwares in different lab machines or students’ computers. Furthermore, professors can use the platform to view the assignments of students online, run their homework online, grade and comment the assignments all in one platform. Compared with submitting a compressed file of source codes to professors, a cloud IDE can enhance teaching efficiency significantly.

Apart from the points mentioned by Kats, Vogelij, Kalleberg and Visser, more features can be implemented to align with typical modern software engineering processes. Similar to team collaboration platform JIRA, user stories can be assigned to different teammates in the organization [11]. In order to further simulate software development, developers can add clients into the platform and send an email to them so that clients can fill in comments during presentation meetup. Developers can also fill in their comments on different user stories all in one page. These user stories are then shown on the IDE panel alongside with the text editor so that developers can focus on the issue and its related comments all on one page.

Java’s shortcomings are well documented. One major obstacle to developer productivity is committing mistakes that the compiler can detect. In one case, the NullPointerException, an error that many Java programmers have encountered, has been called a ‘billion dollar mistake’ by its inventor, Tony Hoarse, who developed its predecessor in Algol W [12]. Although the compiler/interpreter can differentiate between nullable and non-nullable types, the decision to make all object types nullable means that the developer has to spend time doing the compiler/interpreter’s job. Compared with newer programming languages such as Python and X, Java is verbose (using more lines of code to represent the same concept) and often contains boilerplate code. Furthermore, Java is built from Object-Oriented Programming (OOP) concepts, which leads to programming paradigms less dependent on the structure OOP poses to feel unnatural and awkward to write in Java. In response to Java’s shortcomings, different alternative JVM languages, such as Scala and Groovy, have been developed. While they avoid the problems of Java, the need to compile
down to JVM bytecode causes long compilation and linking time, reducing developers’ productivity and desire to use these languages. The goal of this project is to build a new programming language that addresses these shortcomings yet does not hinder developers’ work.

Android is chosen to be implemented in the IDE for the following reasons. First, mobile app usage has significant growth in recent years [14], [15]; as in quarter 2 of 2015, Android has an market share of over 80% in the mobile operating system market [13]. Second, due to the open nature of Android [16], it is more easy to push apps developed by the IDE to Android devices by installing APK files [16], [17], comparing to iOS [16]. Java is also chosen to be implemented because Android use Java as the App's development language [18] and Java is one of the most popular programming language in the world [19].
2. Objectives and Benefits

A cloud-based integrated development environment will be built to facilitate programming in Android, Java and a new programming language. The three guiding objectives the integrated development environment aims to achieve are:

- to facilitate collaborative coding within an organization;
- to make programming lectures more interactive and efficient, and;
- to integrate modern software engineering elements into development processes.

There are many benefits of building a cloud-based IDE platform. Compared with traditional IDE software, such as Eclipse and IntelliJ IDEA, we see the following differences:

- programmers can code anywhere and anytime with Internet access;
- programming projects can be kickstarted instantly without spending time to install any softwares;
- software development can be carried out without hardware constraint, and;
- code can be taught and shared more easily such that more people can be benefited from learning code in a systematic and efficient way.

Furthermore, a new programming language will be implemented in the cloud IDE platform. The benefits of writing a new programming language are:

- to reduce verbosity and boilerplate code;
- to encourage the use of different programming paradigms, and;
- to prevent developers from making mistakes the compiler/interpreter can detect;
- to maintain high level of performance and efficiency.
3. Scope

In this project, we plan to develop the following components:

- a Web-based IDE with coding, compilation and debugging functionality for Android and Java software development, which can support a large number of users;
- a runtime debugging tool that can be installed into users’ Android devices for testing purposes, allowing users to run Java applications on our server and to preview the compiled app in their Android smartphones;
- a new programming language, and its interpreter or compiler, which module developed by this new language can incorporate with the application developed by the IDE and is able to facilitate development
4. Deliverables

1. A web-based IDE with the following features:
   - Coding editor with syntax and API hinting and searching
   - Drag-and-drop GUI editor
   - Seamless collaborative coding
   - Git version control integration
   - Code and application templates
   - Compilation of code in executables or interpretation of code in interpreters
   - Remote delivery of executables to users’ devices
   - Running of Java terminal-based programs with result displayed
   - Capturing logs from running apps into the debugger
   - User account management for teams

2. A backend infrastructure providing virtualization and separate environments for each user

3. An Android Application to receive and execute the application, with debugging message transfer back to the Web IDE

4. A new programming language to create new programming paradigms for building applications, including:
   - Specification
   - Interpreter/compiler
   - Standard runtime environment libraries
   - If time permits, interoperability with Java

5. Approach & Methodology

This project will follow the Agile methodology. Components will be divvied into tasks and assigned priorities. During every sprint, lasting 2 weeks long, tasks will the highest priority will be assigned the each member for completion. At the end of each sprint, the team should be able to develop a usable product and will gather feedback from our supervisor and potential users. With this, the team can re-assign priorities to existing and new tasks (if any) and start a new sprint. Adopting this methodology allows the team to move quickly in face of a shifting goal and direction.
For server-side scripting, it is proposed that the Go language connect the front-end web service to the backend database, as well as contain logic for concurrent development in different browsers. Go, released by Google in 2009, is an emerging programming language renowned for its speed, simplicity and efficiency [20]. According to David Chisnall, the static-typed language has a better performance in concurrency because of its dynamic stack size for threads [21]. Where in Go each thread only requires 4KB of stack, C would need 1MB. Also, channels and locks between Go routines allow thread-to-thread communication. Additionally, its pointer reference, package support, zero initialization and garbage collection is similar to Java features. As such, less time can be spent in learning Go.

Two approaches will be tested in our implementation of concurrent usage by multiple users. The first is a dedicated virtual machine for each user; with siloed computation power, a better performance is likely with the downside of a hard limit of concurrent users supported. The second is a shared virtual machine with separate storage space and processing environment for each user using Docker [22]. Each user would be assigned a pre-configured Docker container, which each container contains necessary developing tools and compiler for Android and Java development [23]. With the use of Docker, a secure environment is provided as each container is separated from the others [23]. There is a higher risk of degraded performance in exchange for the higher flexibility and tolerance of the number of user connection and their amount of usage.
6. Risks, Challenges & Mitigation

As the features required by different users of the IDE are often disparate, scope creep is anticipated. In trying to satisfy multiple user requirements, there is a risk being jack of all trades and master of none. Therefore, focus will be on the key parts of the project that demonstrate the viability of the IDE, and when achieved other features will be slowly rolled out as logically grouped bundles.

The use of newer technologies may cause slippage to our schedule due to time needed to master these skills. There will be constant evaluation of whether too much time has been spent, and if so the possibility of reverting to technologies we know better but possibly less suitable to the task. Once we are back on schedule, then further technical decision can be made.
7. Division of Labour

<table>
<thead>
<tr>
<th>Name</th>
<th>Delivery of</th>
<th>Development of the new programming language</th>
</tr>
</thead>
</table>
| Shum Chi Chung    | • Backend infrastructure, including server and virtual machine set up, database set up, connections between virtual machines and IDE  
                    • Android build system [24] and Java compiler set up on Docker containers[22]  
                    • Connection of IDE to android build system [24] and Java compiler  
                    • Helper app on android for receiving android application packages with debugging tools  
                    • Debugger and logger on web side, receive and display log message from android devices | • Feature design and implementation  
                    • Alternative paradigm support  
                    • Compilation backend targeting Java/JVM bytecode/assembly code/Javascript or interpreter |
| Wong Man Chun     | Delivery of                                                                                     |                                                                                                           |
|                   | • Backend infrastructure, including server logic, database structure  
                    • Frontend development, including UI design, coding editor, syntax hint, API hinting and searching, drag-and-drop GUI editor, collaborative coding, e-learning features, version control, code and application templates |                                                                                                           |
| Poon Stirling Yeu | Development of the new programming language                                                     |                                                                                                           |
# 8. Schedule of Tasks

<table>
<thead>
<tr>
<th>Date</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sep 2015</strong></td>
<td>● Research literature and existing solutions&lt;br&gt;● Collect user stories&lt;br&gt;● Learn required technologies and skills&lt;br&gt;● Create project plan and website</td>
</tr>
<tr>
<td><strong>4 Oct 2015</strong></td>
<td>Delivery of&lt;br&gt;1. Detailed project plan&lt;br&gt;2. Project web page</td>
</tr>
<tr>
<td><strong>Early Oct 2015</strong></td>
<td>● infrastructure and server set up</td>
</tr>
<tr>
<td><strong>Late Oct 2015</strong></td>
<td>● First prototype of web IDE&lt;br&gt;○ Web framework, coding editor, code and application templates&lt;br&gt;○ Java code editing, compilation and execution&lt;br&gt;● First prototype of new programming language</td>
</tr>
<tr>
<td><strong>Mid-late Nov 2015</strong></td>
<td>● Second prototype of web IDE&lt;br&gt;○ Collaborative coding, version control&lt;br&gt;○ Android text based code editing and compilation and execution on Android devices&lt;br&gt;● Second prototype of new programming language</td>
</tr>
<tr>
<td><strong>Late Nov 2015</strong></td>
<td>● Third prototype of web IDE&lt;br&gt;○ Syntax highlighting, debugging support&lt;br&gt;○ Initial new programming language integration</td>
</tr>
<tr>
<td><strong>Late December 2015</strong></td>
<td>● Fourth prototype of web IDE&lt;br&gt;○ Command-line display, run output display&lt;br&gt;○ Syntax hint, simple code error detection&lt;br&gt;● Third prototype of new programming language</td>
</tr>
<tr>
<td><strong>Early January 2016</strong></td>
<td>● Presentation of the first rollout of IDE</td>
</tr>
<tr>
<td>Month</td>
<td>Events</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>February 2016</td>
<td>● Fifth prototype of web IDE</td>
</tr>
<tr>
<td></td>
<td>○ e-learning features</td>
</tr>
<tr>
<td></td>
<td>○ Second new programming language integration</td>
</tr>
<tr>
<td>March 2016</td>
<td>● Sixth prototype of web IDE</td>
</tr>
<tr>
<td></td>
<td>○ Project management, issue tracking</td>
</tr>
<tr>
<td></td>
<td>○ Initial new programming language integration</td>
</tr>
<tr>
<td>April 2016</td>
<td>● Second rollout of web IDE</td>
</tr>
</tbody>
</table>

Note: the schedule after Jan 2016 is subject to change based on comments from our mid-term review.
9. Conclusion

Hearing from different users of other IDEs, we are excited to build the next-generation IDE. By coding at any time even without their laptops, users can focus on developing their application instead of spending time to install software and additional packages. Furthermore, the team integration features of our IDE will allow them to collaborate more closely with team members and lead to higher team efficiency. We look forward to seeing our users start using our product and telling us what they think.

We would like to thank our teachers and advisors, Dr. T.W. Chim, Dr. Jon Hui and Dr. Bruno C. d. S. Oliveira for their advice and guidance.
10. References


