Final Year Project

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

Final Report

Shum Chi Chung
3035070236

(in collaboration with Wong Man Chun and Poon Stirling Yeu)

Supervised by
Dr. Chim Tat Wing

April 17, 2016
Contents

1 Introduction ................................................. 2

2 Objective and benefit ...................................... 3

3 Literature review and existing works ....................... 5

4 Scope .................................................................. 6

5 Design and Procedure ....................................... 7
  5.1 Platform setup ................................................ 7
  5.2 User account structure ..................................... 7
  5.3 Web IDE Front-end ......................................... 8
    5.3.1 Login page ............................................... 8
    5.3.2 Catalog Page ............................................ 8
    5.3.3 Editing Page ............................................ 8
  5.4 Web based IDE back-end ................................... 10
    5.4.1 Use of Go language .................................... 10
    5.4.2 Database ............................................... 10
    5.4.3 Compiling and executing environment ................. 10
  5.5 Application development workflow ....................... 11
    5.5.1 Java applications ...................................... 11
    5.5.2 Android applications .................................. 11
  5.6 User Interface and User Experience design ............... 14
    5.6.1 Web IDE ................................................ 14
  5.7 New Programming Language ............................... 22

6 Deliverables ...................................................... 25

7 Division of Labour ............................................. 26

8 Results and Challenges ....................................... 27

9 Future works .................................................... 28

10 Conclusion ...................................................... 30
Chapter 1

Introduction

In modern software development, an Integrated Development Environment (IDE) is a commonly used software which include a set of tools to facilitate software development. Common example of tools in an IDE including code editor, which may contains features such as syntax highlighting and hints to improve readability of code and enhance speed of writing codes. Another example is debugger which may display debug message in a way that making errors easily traceable by programmers. The purpose of this project is to build a next generation IDE by redesigning some interfaces and features of tradition IDE and adding new and innovative features and tools so as to provide a solution which increase productivity during a software development process. This project includes two main parts. One part is the Web-based IDE for Java and Android programming with a set of new features and another part is a new programming language Java+ which is usable in the Web-based IDE which is concise and neat translation of the Java language. This is a group project consist of three members, includes the author of this paper Shum Chi Chung and another two members Wong Man Chun and Poon Stirling Yeu. The responsibility of each member will be discussed in the Chapter of Division of Labour.
Chapter 2

Objective and benefit

In this project, the main theme is to investigate into features that are able to enhance productivity of developer. In this sense, a Web-based IDE is developed. An web-based IDE does not require users to install any additional software other then a web browser[24]. This help users to save storage space on their computers. This also saves the time for users to install or update the software, tools, or compiler provide in an IDE on their personal computer[24]. As a Web-based IDE is access through a web browser with an internet connection, thus any computing device with a web browser can access to the Web-based IDE[24]. Traditional IDE, which may required install software on personal computers, may not have a version that supports mobile device. By using a web-based IDE, developer can access to the IDE with a mobile device[24]. This allow programmer to work on their projects with any device at anywhere. Beside, in tradition IDE that are installed in a personal computer, the compilation work is done on the users’ personal computer. The speed of the compilation may depends on the computation power of that personal computer. If developers found the compilation is slow on their computers, it may require an hardware update which cost money. In a web-based IDE, all the compilation work is done on the server, which is independent to the computer which the developers use to access the web-based IDE. This eliminates the dependence of compilation time to computing power of users’ computer and reduce the cost for users to upgrade their computer for software development.

This Web-based IDE is named BuildApp. The name simply tells this is a web platform that help people to build their own software applications.

One of the main features of the Web-based IDE in this project is collaborative editing. Sometimes multiple number of developers are working together on the same software project and even working on the same file[24]. Collaborative editing is introduced to allow developers to work on the same file together concurrently[24]. In traditional way, multiple developers may work on their own 'branch’ of the project and require a merge after each one finishes[24]. Collaborative editing ease the need to merge, as changes are applied concurrently[24]. This saves time when multiple developers are working together in a rapid development cycle[24].
Android application programming is chosen to be implemented as a kind of project that can be developed in the IDE due to the following reasons. Mobile application usage has experienced significant growth in recent years [23, 7]; as of the second quarter of 2015, Android has had an market share of over 80% in the mobile operating system market [18]. Second, due to the open nature of Android [17], it is easier to push applications developed by the web-based IDE to Android devices by installing APK files [17, 16] when compared to the iOS platform [17]. Java is also chosen to be implemented because Android uses Java as the application’s development language [6] and Java is one of the most popular programming language in the world [29].

As Android Programming is included in the Web-based IDE in this Project, some features that is possible to enhance productivity is considered to be implemented. One of those features is media recording. With using the android.media.projection Application Programming Interface(API) in Android 5.0[3], users can record the screen output of their Android devices[3]. Another feature is remote device deployment, which allow deployment of Android Application to Android device with internet connection only. When develop wants to test the application on Android device, in tradition way, the application is pushed from the IDE to the mobile device though USB cable via the Android Debug Bridge(ADB)[4]. It requires plugging the USB cable or a number of set up to achieve wireless deploying and debugging[4]. A remote deployment mechanisms is to provide a fast track for simple deployment of application to Android devices. This can also be used on clients’ device where developers can push the latest completed app into client devices immediately and obtain feedback from client.
Chapter 3

Literature review and existing works

There are number of traditional IDE in the market such as eclipse [15], Android Studio[2]. Although they provided a huge number of mature tools for software development[2, 15], they are required to be installed on users computer and taking cost to time and storage space[2, 15]. There are also a number of Web-based IDE in the market. Including Codenvy[12] and CodeAnywhere[11]. Both provided with a numerous number function[11, 12]. However, some functions like remote Android device deployment and media recording is missing as those Web-based IDE may not target on Android Programming.
Chapter 4

Scope

As this is a one year project, due to time limitation, only some basic features of an IDE as well as some new feature are developed.

The below is a list of feature developed:

- A new user interface
- A collaborative editor
- Project File listing
- Compiling/Building of Java, Android and new programming language Java+
- Executing of simple output-only Java program
- Wireless deployment of Android application to real devices
- Media recording in Android devices
- Integration of new Programming language Java+ into the IDE
- Issue tracking functions
Chapter 5

Design and Procedure

This chapter will describe the design and implementation of the whole system developed in this project.

5.1 Platform setup

A server and an IP address 147.8.138.3 has been provided by the University of Hong Kong, Faculty of Engineering, Department of Computer Science (the Department). Debian operating system was installed in the server machine.

A domain name buildapp.xyz is purchased for easy access to the website. The domain name is linked to the protection service by the company Cloudflare which provides Domain Name System (DNS) service, Secure Sockets Layer (SSL) services and other security service which provides more protections to the website and server [10].

Only limited number of ports are opened by the Department to public access to the server. In order to maximize the number of service provided by the service, Nginx is used as the web server and proxy passes to service opened at different port at localhost are created and linked to different sub-domains. The sub-domain code.buildapp.xyz is chosen to link to the Web IDE.

5.2 User account structure

In the IDE, an organization is the main structure. Each user must belong to an organization. Projects are also belongs to an organization. Thus, only member users of an organization can access to the projects of that organization.
5.3 Web IDE Front-end

The front-end of the web IDE include three main interface.

5.3.1 Login page

The login page is the entry point of the IDE. User login to the IDE with their email and password, as shown in Figure 5.1.

Figure 5.1: Login page

5.3.2 Catalog Page

The catalog page allows users to manage the project of their organization. User can view the list of projects in this page, as well as creating new projects. Users can also create new user account for their organization to login to the IDE and access the projects of the organization.

5.3.3 Editing Page

When user click a project entry in the project list of the catalog page, that project will be opened in the editing page. This page provides the main features of the IDE, including
collaborative editor for files in the project. When multiple developers open the same file in the same project, they can see the changes made by others immediately. The collaborative editor used the technology operation transformation which was proposed by Ellis and Gibbs in 1989[9]. The editing page also includes the function of compiling and running or deploy of the project. This page also includes the management of Android devices and the media files collected from device and the issue tracking panel.
5.4  Web based IDE back-end

This section describes the technology used by the back-end server of the IDE.

5.4.1  Use of Go language

Go language, which is developed by a team of Google developers and some other contributors,[28] is chosen as the language to implement the server side program.

5.4.2  Database

MongoDB has been chosen as the database system. Reasons are as follows. From Seguin, there is no need to define a fixed schema for the collections in the database and this provide more flexibility[22], thus the data structure in the system can be change rapidly only at the program but no need to change any settings at the database side. Besides, with this flexibility on the schema, it works better with Object Oriented Programming, which objects could be saved directly to database after serialization[22], even object structure has been modified.

5.4.3  Compiling and executing environment

Docker[14] is implemented to provide an compilation and execution environment for users. Each organization is provided with a preconfigured Docker container which contains necessary and most updated developing tools and compilers for Android and Java development includes the Android Software Development Kit and the Java Development Kit. The Docker Container allow the possibility to provide a fully functional terminal as in a personal computer, allowing access to the underlying Linux Operating System of the Docker container. Users are allow to manipulate the docker container as they own computer. This provides the flexibility of installing additional software, compiler or tools inside the docker container to assist their development work other then just compiling Java or Android project. Thus, developers are provided with a highly customizable development environment. However due to time limitation in the project, this feature will leave to future work.
5.5 Application development workflow

5.5.1 Java applications

To develop a Java application, users need to create a Java project in the catalog page. The flow of developing a Java application is shown in figure 5.2. Users can write their code on the front-end of IDE collaboratively. After users finish writing the code, they can start the compilation of the source code files of the project, which is carried out inside a Docker container assigned to the organization. After the compilation has finished, users can execute the compilation result. The execution output is sent back from docker container to users’ web browser and show at the ‘Running Task’ panel at the web-IDE interface.

![Java Project development diagram](image)

Figure 5.2: Java application development work-flow

5.5.2 Android applications

For Android application development as shown in figure 5.3, after users create an Android project, they can write code collaboratively. After clicking the ‘Compile’ button on the interface, the Android project is then built inside the Docker container assigned to the organization. The application package would be generated.

In order to deploy Android Application projects for testing and debugging, the IDE supports deployment of Android application on ‘real’ devices in wireless through the internet. At First, developers need to install the Agent application named ’BuildApp Agent’ on their devices and login the BuildApp Agent with they account on the IDE,
2. Android Application Development

Figure 5.3: Android application development work-flow, the collection of debug message will leave to future work

shown in 5.4, and 5.5. When the developers want to test the successfully built application, they should first navigate to the 'Device' panel by clicking the 'Device' button on the left navigation panel. After that, the 'Device' panel will be shown on the panel below the code editor where they can select the device to deploy. After they clicked the 'Deploy' button, the server will send a command to the BuildApp Agent on the selected device(s), which then will automatically download the application package upon successful receiving the command. After the download completed, the application installer will be triggered. Due to the security policy enforced by the android system, the application installer will as the developer to accept the permissions required by their own application before the application can be installed, shown in figure 5.6. Upon successful installation of the application, developer can start their testing. Developers can also use the floating tools provided by the BuildApp Agent to record video and capture a image from the device screen, these media will be uploaded to the IDE immediate upon successful recording.

In order to enhance the efficiency in software development, BuildApp also provides features that helps streamline software development workflows. For Android projects, BuildApp can deliver the testing app not only to developers’ device but also to clients devices by simply installing the BuildApp Agent onto client devices. BuildApp Agent also allows clients to provide feedback or issue reporting directly from the device to the developers. The client needs to install and sign in the BuildApp Agent with a client account or with an invitation code from developers. The developers can push the latest version of the application to the client immediately and the client can try and test the application. Clients wants can give feedback or reporting any software bugs by using the create issue function and they can attach media file by taking screenshots or screen recordings using the floating tools of the BuildApp Agent. The developers can then im-
prove the application by view client’s request listed in the issue tracking panel. However, due to time limitation, the functionality of client usage on the agent app will be left to future work.
5.6 User Interface and User Experience design

This section describes the User Interface (UI) and User Experience (UX) design of different components.

5.6.1 Web IDE

When implementing the user interface of the web IDE, bootstrap is used which provided the basic framework of HTML, CSS, Javascript. This facilitates the design of the UI as bootstrap provided a large number of components. In some of the pages, such as the login page, the bootstrap theme AdminLTE is applied. Some of the design ideas are inspired by the material design of Android.

Catalog page

In the catalog page, different sections are separated by tabs on the top of the page. When users navigate to other sections, such as changing from view the project list to crate project, the user click on the tab and navigate to the desired section. The navigate is completed by calling respective Javascript functions at the browser. This approach removed the loading latency as only the data in the new section would be loaded, instead of loading the whole page again. Beside, transition animation, using Animate.css, is applied during the navigation of sections. This also makes transition smooth and makes the IDE more fancy. Figure 5.7 shows the project list section and Figure 5.8 shows the create project section.

Editing Page

The editing page, as shown in Figure 5.9 includes a number of panels and elements. The elements include, the top function bar, the left navigation panel, the code editing...
There will also be notifications showing the progress of compilation and deployment, as shown in Figure 5.10 and Figure 5.11. The top function bar provides buttons for some global functions of the IDE, such as triggering compilation of the project. The top function bar also includes the selection of theme of the editor panel, as well as navigation buttons including the back to catalog and logout button.

The left navigation panel as shown in Figure 5.12 provided a number of buttons to access different panel in the IDE. The ‘Files’ button gives access to the tree view of the files in the project, on the same panel. Users can view the folder and file structure of the project. When users click on a folder, the folder will be expended and showing the content inside the folder. When users click on the file, the file will be opened in the code editor. When right click on the folders or files, action panel will appear which the user can perform action includes adding new file, folder or deleting file. Below the ‘Files’ button, there are a number of buttons includes the ‘Running Task’, ‘Terminal’ and ‘Devices’. When clicking one of these buttons, the multi-function panel below the code editor will be changed to the respective panel, and showing the respective contents. The
design of this panel came from panel on the mac desktop application of the an application 'AirDroid'[27].
Upon opening a file in the 'File' panel on the left navigation panel, the file list will be shown on the code editor. Different developers may have different preference on the style and theme on the editor. Therefore, a theme selection is provided on the top function bar. Six themes are provided for selection when clicking the drop down menu 'Editor Theme' at the top function bar, as shown in Figure 5.13. Three light color theme and three dark color theme are provided for selection to meet the personal need of different developers. Example of dark theme is shown at Figure 5.14 and a light theme shown in Figure 5.15. As the editor is a collaborative editor, which may be more then one person are editing the same file. Thus, there is a tag showing the editing position of other developer on the code editor as shown in Figure 5.15.

The multi-function panel under the code editor shows different content according to the selection from the left navigation panel. Different panel will use different colors in the heading to allow users can distinguish them easily. The 'Running Task' shows the last result from the compilation task of the project. The 'Terminal' shows a terminal provide a shell access to the docker container assigned to the organization, which is not implemented due to limitation of time. The 'Devices' panel list the Android devices registered as testing devices of the organization.

In the 'Devices' panel, it shows a device entry which includes the screen size and operating system version, the product image of the device if found, the device name to differentiate different devices, as shown in Figure 5.16. With showing screen size and
operating system version, this makes the developers no need to remember this kind of detail of each device and easier to figure out possible problem when running the Android application on different device due to their difference in screen size or operating system version.

There is a 'Detail' button below each device entry. When it is clicked, a popover will appear and it will show more detail information about the device includes the screen resolution and Android API Level, as shown in Figure 5.14. Inside the popover, it also contain a number of buttons. The first button is the Deploy button, when it is clicked, the latest built of the Android application will be push to the respective Android device. The 'Device' panel also provides a way to achieve a large scale deployment. The users can select multiple devices by clicking the device image or icon. The selected devices will be highlighted in grey color. To deploy, user should click the 'Deploy' button at the heading of the 'Device' panel and the deploy request will be sent to the select devices.
In each device entry, there is a status light at the bottom right corner, as shown in Figure 5.16. This light shows current status of the deployment action. A yellow colored light with arrow up icon indicates the deploy request is sending to the device. A blue colored light with an arrow down from cloud icon indicates that the device received the deployment request and downloading the application from the server. A green colored light with a tick indicates the download of the application has been completed and user
can install the application into that device. A red colored cross shows error occurs during the deployment request.

In the popover when the 'Detail' button is clicked, there is a 'Media' button next to the 'Deploy' button. When the 'Media' button is clicked, the 'Media' panel on the right of the multi-function panel will show the list of media recorded from the device, as shown in Figure 5.18. When clicking an entry of the media list, the media file, either an image or a video, will be opened above the media panel for viewing, as shown in Figure 5.19.
The 'issue' panel on the left shows the issues found at the project, with the status of whether they are fixed, is fixing by someone or waiting for someone to fix. Clicking new issue will allow users to add an issues, as shown in Figure 5.20.

Figure 5.20: Creating a new issue of the project

**BuildApp Agent**

Upon user login in the app, the developers can start deploying the application by finding the device at the 'Device' panel. The BuildApp Agent use the system installer of that device to ask user to install the application.

After the first time login to the app, subsequent launching of the Buildapp Agent in the device will trigger the Floating tools. This floating tools is inspired from the 'Chat heads' of the Facebook Messenger application on Android[25]. The floating tools always float on top of other applications, except for system dialog. This allows the users to access the tools while they are testing their own applications. By clicking the float BuildApp icon on the screen, the tool set can be expanded or collapsed. An expended floating tools is shown in Figure 5.21. The user can then use the tools according to their needs. Notification will appears during screen recordings and screen capture completed. Figure 5.22 shows a notification that indicates it is recording the screen. The media recording will be upload to IDE immediate after recording success. There is also notification showing the media is uploading, as shown in Figure 5.23.
5.7 New Programming Language

A new programming language Java+ is implemented using a framework named xtext.
Figure 5.22: An Notification indicates the tools is recording the screen
Figure 5.23: A Notification showing the progress of uploading a media file.
Chapter 6
Deliverables

The following are the deliverables in this project

- A Web-based IDE
  - The Front-end user interface
  - The back-end server side program
  - The back-end Docker container configuration
- A agent Android application - BuildApp agent
- A new programming language Java+
Chapter 7

Division of Labour

This is a group project consist of three members. Members include the author of this paper, Shum Chi Chung and the other two member Wong Man Chun and Poon Stirling Yeu.

Shum Chi Chung works mainly on the front-end IDE design and implementation, such as the styling and displaying or positioning, some the the server side program such as the logic to access docker container, start a compilation job, application deployment to Android devices, the docker container configuration and the development of the BuildApp Agent.

Work of Wong Man Chun including the implementation on file structure and display of folder tree structure at front-end, project listing and project creation at front-end and back-end, the collaborative editor and the issue tracking function, as well as the users and organization structures and the implementation of user login and session.

Poon Stirling Yeu is responsible for the work on the new programming language Java+ and its integration into the IDE
Chapter 8

Results and Challenges

This project was taken to join the Hong Kong ICT Awards 2016: Best Student Innovation (Tertiary or above) and won the Certificate of Merit. At the time joining the award, some useful feedback was also obtained from the Judges. Some judges raise a concern on the collaborative editing features. When multiple developers are working together, once a developer finished his work, he may wants to test on his part. However, in the current implementation, if someone is still editing the file, the developer may not able to build the project successfully and the file may contains incomplete codes from other developers. This may affect efficiency in some way. The solution to this issue will be discussed in the future work.

There are also some other challenges in this project. In the BuildApp Agent, due to some security policy enforced by the Android Operating System, some of the features are not able to completed so easily and dropped from the scope due to time limitation. For example, it is planned to implement the remote logcat reader at the IDE, which the BuildApp Agent collects log messages from devices to send to server and allow developers to read for debugging purpose. However, due to the security policy enforced by the Android Operating System, applications only allow to read their own log[5]. That is, the BuildApp Agent is only allowed to read the logs generated by the BuildApp Agent itself[5]. This make the logcat reader useless as users need to read the logs generated from their own application. According to Boyd, by run the logcat process on Android device after using the ‘su’ command on rooted devices, the log can be read[26]. Due to time limitation, this approach is not bring to current implementation. Also, this implementation only work on rooted devices and requires developer to do more work on their devices and have a possibility to break the warranty of the devices[21].

Beside, in the Screen recording and capturing feature on the Agent App, the android.media.projection Application Programming Interface(API)[3], which is new in Android 5.0[3] is used for the implementation. Therefore, in device with Android operating system version lower then 5.0, this function is not workable. In spite of this, as the project is target to build a next generation IDE for the future, as long as this API is kept in future version of the Android operating system, this tools can still be used to improve productivity and efficiency of Android Application Development.
Chapter 9

Future works

To address concern on the collaborative editing features at the Hong Kong ICT Awards 2016, as stated in the chapter of ‘Results and Challenges’, some possible solutions have been come up. Goldman, Little and Miller in 2011 comes up an algorithm which allows running of program only on the code that has no error[24]. In this way, developer no need to wait for the others the complete the files and can test the project.

Beside, for users that are used to using version control system such as git or svn to achieve collaboration, it may not easy to for them to switch to using a collaborative editor. Therefore, version control system should also be implemented to provide choices to developers. Possibly collaborative editing is enabled first and allow users to switch to use version control system. Users can even use both collaborative editing and version control system together to obtain the benefit from both approach.

Other future works include completing the features that has not completed in this project due to time limitation. Including a logcat reader, for rooted device[26] as well as the issue reporting mechanism from BuildApp Agent, and the client version of Buildapp Agent.

Due to the security policy enforced by Android operating system, many debugging feature on Android Device is only workable when the device is plugged-in to a computer and access those features using the Android Debug Bridge(ADB)[4]. In order to provide this set of features, one possible solution is to make a plug-in of the web browser. This solution is came from the Google Chrome Browser plug-in ChromeADB in the market[20], which able to obtain information from device connected to the computer through the ADB[20]. In order to meet the features and need of the Web-based IDE. There is a need to develop a similar plug-in which the browser plug-in allows the communication between devices and the web-based IDE. The web IDE access information or executing programs in the Android Device through the plug-in. With this approach, log messages can be obtained from normal non-rooted devices and send to the server.

At the start of the project, there is a hope to develop an IDE for professional use in production environment. In the future, a lot of works has to be completed to achieve this
goal. The works includes implementing a complete set of feature in existing IDE, obtain users feedback from large scale testing, etc.
Chapter 10

Conclusion

In this project, a prototype of an next generation IDE is worked out. Due to time limitation and lack behind of schedule, some features was not able to be completed. Although concern has been raised to some of the features, the project has successfully bring out innovation idea and features for building the next generation IDE. Some features of an IDE bought out in this project has received positive feedback and is reflected on winning the Certificate of Merit in the Hong Kong ICT Awards 2016. As this project only implemented some of the feature included in an IDE, there is still a long way to implement a production IDE and publish to the market.
Bibliography


