Academic Year 2024-2025 Semester 1

50002870006 - Java Enterprise Application Development

Assignment A

Requirements

Each team of candidates is required to design and implement an IntelliJ IDEA plugin that supports automatic and fine-grained source code versioning functionalities and features. The assessment will be based on both the presentation and submission. All teams will be randomly divided into three groups (namely A, B, and C), while each group is subject to different requirements as below:

Group	Requirements	Submission Due	Presentation Date and Time
Group A	F0	11.59pm, 8 October 2024	1.30-4.15pm, 9 October 2024
Group B	F0 + F1	11.59pm, 15 October 2024	1.30-4.15pm, 16 October 2024
Group C	F0 + F2	11.59pm, 22 October 2024	1.30-4.15pm, 23 October 2024

F0: Basic Functionalities (For All Groups)

During the programming process, the plugin automatically tracks and analyzes fine-grained source code modifications made by the programmer, and saves source code versions with a ***reasonable*** policy. The plugin must also provide a user-friendly interface, allowing the programmer to easily view any of the saved versions of the source code.

Hint: You may refer to Tencent Docs (or similar products) for the concept of fine-grained document versioning, but please note that a programming environment deals with multiple source code files rather than a single document.

F1: Extended Feature 1 (For Group B Only)

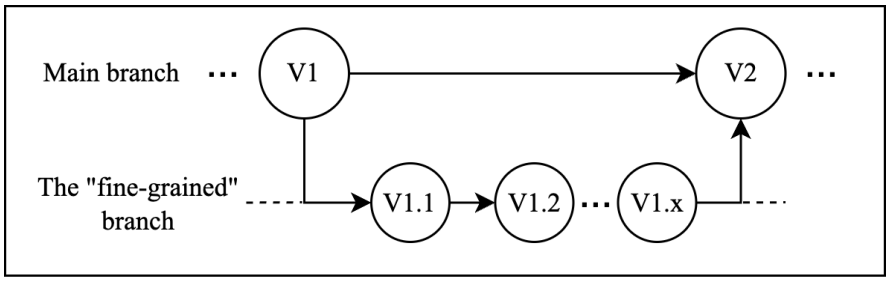
Following the above requirement, when a saved source code is being viewed, its differences from the current source code copy can be intuitively observed. Additionally, the programmer should be able to easily revert the current code to any of the saved versions with minimal effort.

F2: Extended Feature 2 (For Group C Only)

This plugin also provides a button for initiating Git commits. By clicking it, the plugin creates a new branch from the current one, tracking fine-grained changes as individual commits. At the same time, the series of fine-grained changes are treated as one single commit and applied on the original

branch, keeping the commit history clean. (In case the project is not yet connected to any Git repository, the plugin will firstly initialize one.)

The abovementioned fine-grained commit mechanism can be intuitively illustrated as the figure below. The main branch goes through versions V1 and V2 as single consolidated commits, while on the “fine-grained” branch, V1 has been broken into smaller and incremental changes as V1.1, V1.2 and more, which correspond to the fine-grained source code versions as described in F0.



References

1. IntelliJ IDEA plugin development: https://plugins.jetbrains.com/docs/intellij/developing-plugins.html
2. If you have any question on IntelliJ IDEA API and plugin development, you may wish to ask questions on https://intellij-support.jetbrains.com/hc/en-us/community/topics/200366979-IntelliJ-IDEA-Open-API-and-Plugin-Development. There will be volunteers to answer your question.
3. Tencent Docs: https://docs.qq.com
4. IntelliJ IDEA’s *Local History* feature: https://www.jetbrains.com/help/idea/local-history.html
5. The *diff* algorithm: https://en.wikipedia.org/wiki/Diff
6. Git: https://git-scm.com

Submission

Each team is required to submit a package consisting of the following materials:

a) The complete set of source code;

b) One project report that covers the following contents in the given order:

   1) Full names and matriculation numbers of all team members, contact number, email address, and other particulars if any;
   2) The complete list of system functionalities and features implemented;
   3) A simple user manual (including UI samples and descriptions);
   4) System architecture and components, general workflow, versioning policy and design rationales, major data storage and structures (e.g. how and where the fine-

grained versions are saved), key algorithms, major supporting techniques, and efficiency considerations;

5) Other technical details and/or information, if any.

Note: The report should be written in English; file format for submission: PDF.

c) The presentation slides, with the first slide presenting the full names and matriculation numbers of all team members.

Note: The slides should be composed in English; file format for submission: PDF.

Please place the source code (a) and the documents (b and c) into different folders, compress them into one ZIP file, and submit it via *Canvas*.

Presentation

Each team is required to present the project within 20 minutes, followed by a Q&A session. The presentation should be delivered in English. Candidates are required to:

a) Explicitly demonstrate all implemented functionalities and features;

b) Present the design of system architecture and components, general workflow, versioning policy and design rationales, major data storage and structures, key algorithms, major supporting techniques, and efficiency considerations, as well as other issues that are worth discussing;

c) Present the mechanisms, methods and skills of collaboration among team members;

d) Discuss important issues involved during the design and implementation, as well as the corresponding solutions.

Grading Criteria

Grading Criterion	Weight
Part A: Presentation	
System functionalities and features	35%
Technical design	35%
Presentation quality	10%
Part B: Submission	
Source code quality	10%
Documentation quality	10%

THE END