Researchers have shown that over 80% of software development costs are spent on maintenance. It has also been shown that over 50% of the effort spent in code maintenance is dedicated to code comprehension. As modern software systems grow larger and larger, this problem continues to grow. Nowadays, systems span several software platforms and languages. For example, consider a system with Android, iOS, and web (javascript) frontends with PHP, Ruby, Java, C#, Go, and Node.js as backends. For most projects, documentation focuses on Java-doc like comments for classes and their methods and does not shed light on the parts needed to implement specific use-cases. In particular, these documents do not capture how the implementation spans these platforms and languages.

To address this problem, we present Panorama, an Eclipse plugin, a semi-automated tool, designed to detect and to document with an orthogonal (to java-docs) focus on the parts of code needed to implement specific use-cases. We show results of an experimental evaluation to show the effectiveness of this tool in helping developers perform code maintenance. We also present preliminary results of a tool designed to extract or copy the identified parts of code for use-cases to help in code reuse of open source software.