Professor and Class,

The Java EE platform is a project that offers a selection of high-level APIs that provide in-demand features that allow developers to produce scalable, transactional, and cost-effective applications. As an example, the Java EE platform simplifies dependency injection from a tedious, XML-formatted approach to simple in-line annotations that are handled by the Java EE system at runtime.

Looking at Java EE versions 8 and 7:

**Version 8**:

Java EE 8 introduces a new Web Security API, which provides enhanced support for authorization and authentication in web modules, as well as providing access to various “identity stores". This has the obvious benefit of allowing programmers to more easily implement authentication schemes into their applications, which would otherwise take more paid hours or more needless complexity in code. This update included more quality-of-life improvements like the one aforementioned. Nothing stood out as “groundbreaking” with this release.

**Version 7:**

This Java EE version released many other quality-of-life improvements, which of course, preceded the version 8 release. These included greater inclusions of annotation-driven configuration capabilities, which have the tendency to be far easier to maintain and edit than XML-based configuration files. There was also added support for POJOs and simplified packaging features. This version had an extraordinarily short release summary.

At this point, it appears that the core features of the Java EE platform have all been satisfied. From here out, many improvements appear to target the developer experience and improve the developer's role in project maintenance.

Refs:

<https://javaee.github.io/tutorial/overview001.html>

<https://www.oracle.com/technetwork/java/javaee/documentation/ee8-release-notes-3894362.html#Whats_New>

<https://docs.oracle.com/javaee/7/tutorial/overview001.htm>

Regards,

Daniel