**ISYS3001 – Assignment 1 exercise**

Remember that this is a public repository - your changes could be seen by anyone who looks!

Add some comments about Version management outside this border, or just add some text so there is a change to this file.

Remember that your GitHub user ID must be submitted in your assignment report!

Once you’ve changed follow the next step in your assignment task.

…

When talking about version management, it usually refers to managing different versions or versions of a system in a software development project. Versioning helps to track the evolution of a project, coordinate multi-person collaboration, roll back to previous states, and protect the project from data loss or errors. Here are some notes on version management:

Version Control System (VCS) : The foundation of version management is the use of a version control system. These systems allow developers to track changes to files and code and record a detailed history of those changes.

Local version control: Local version control refers to the use of simple version management tools, such as file copies or directory copies, on the developer's local computer to keep track of different versions of files.

Centralized Version Control (CVS) : In a centralized version control system, the history of all project files is stored on a single central server. Developers need to communicate with the server to get or submit updates.

Distributed Version Control (DVCS) : A distributed version control system not only stores project history on a central server, but also stores a full copy of the history locally for each developer. This allows developers to work offline and make it easier to branch and merge code.

Commit: A commit is a step in saving a file or code change to a version control system. Each commit creates a new version.

Branch: A branch is a separate copy of a code base that allows developers to experiment or work without affecting the main development line. Branches are often used to develop new features or fix bugs.

Merge: Merge is the process of merging changes from two or more branches into a single branch. This often requires conflict resolution to ensure that the merged code remains consistent.

Tag: A tag is a meaningful name or tag used to identify a particular version. They are often used to release stable versions so that they can be easily retrieved later.

Conflict: A conflict occurs when two or more developers try to merge incompatible changes. Resolving conflicts requires manual handling or choosing how to merge changes.

Rollback: A rollback is the operation of restoring a project to its previous state, usually to fix errors or cancel unwanted changes.

Continuous integration (CI): Continuous integration is a development practice where code is frequently integrated into a shared repository and then automatically built and tested to ensure the quality and stability of the code.

Version management tools: There are many version management tools to choose from, including Git, Subversion, Mercurial, and more. The choice of tools should be based on the needs of the project and the preferences of the team.

Version management plays a critical role in software development, helping teams collaborate, track project progress, and ensure code stability. Choosing the right version management strategy and tools for your project and team is one of the most important decisions in the development process.