Version Control Scheme

for

Online PCR Tests Booking Platform

at The Department of Electrical and Computer Engineering,

The University of the West Indies

St Augustine Campus

Trinidad

Version 1.0

Prepared by:

Fahad Hosein

Team Gryffindor

30/09/2021

# Document Control

Title: Version Control Scheme

Version: 1.0

Date: 30/09/2021

Author(s): Fahad Hosein

## Document Signoff



|  |  |  |  |
| --- | --- | --- | --- |
| Nature of Signoff | Person | Signature with Date | Role |
| Author      Reviewees | Fahad Hosein      Tahj Ramudith |  | Configuration Manager  Product Manager |

## Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Change Details |
| 21 Sept 2021 | 0.0 | Fahad Hosein | * Drafted introduction * Drafted application version control scheme * Drafted document version control scheme |
| 30 Sept 2021 | 1.0 | Fahad Hosein | * Completed introduction * Completed application version control section * Completed document control section * Revised application version control scheme |

# Introduction

Version control involves the tracking and managing of changes to software code and documents. Each modification to the documents or application code must be logged in changelogs/change-boards. This is vital for the development of software and documents since collaborative work can often lead to conflicts in code and documents. Furthermore, it mitigates the risk that if errors develop in either the code, the project can be easily reverted to the last known working version. It also facilitates easier error detection when all the changes to the software are logged.

# Software Version Control

## Version Control System

The version control system implemented is distributed version control. The distributed version control system is used for the development of the application since it is faster and is less prone to errors, i.e., any errors made to the code can be reverted since the changes are staged. This system involves the cloning of the repository to a local machine. After the repository is cloned, the developer fetches recent changes from the repository, this process is known as “pulling”. After the developer makes any changes to the most recent version of the code, the changes are committed (or staged). This creates a snapshot of the repository with the changes made, such that it can be reverted if needed. After the changes are committed, they are then applied to the server’s copy of the repository. Figure 1 shows a simplified diagram of the distributed version control system.

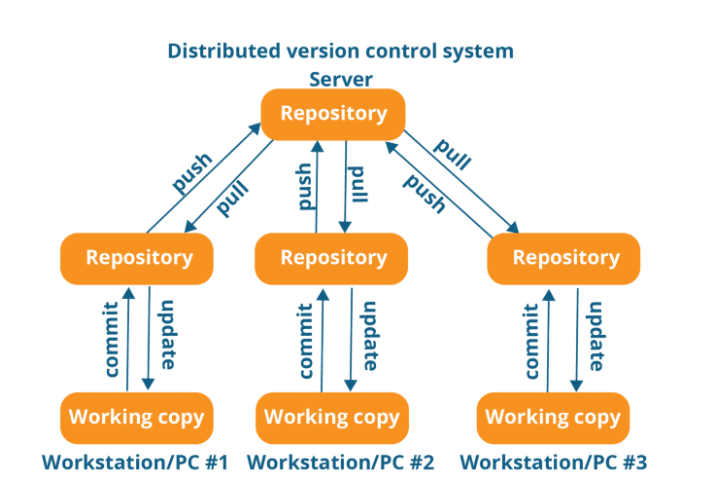


Figure 1: Distributed Version Control System Diagram (Lubański 2019)

The benefits of this system are that it is faster and more reliable than centralized version control, it allows for easy branching and merging and the repository can be cloned on a local system, i.e., the developer does not need a constant connection to the server.

## Version Control Tool

The version control tool chosen which implements the distributed version control system is git. GitHub was used for the server-side hosting of the repository. Git is used for tracking and staging commits to code, and facilitates pull and push requests by users to the server repository hosted on GitHub.

## Version Control Scheme

**Version Control Method:** Centralized

**Notation:** Version\_A.B

**Parameters:**

1. A.B: Version Number
2. **A:** Incremented each time the software is updated to a complete, finalised version. The initial value is 0. Each time the A value is incremented, the B value is reset to 0.
3. **B:** Incremented each time minor updates are made to the code. The initial value is 0.

**Example:** Version\_1.0

# Document Version Control

## Version Control System

Version control of the documents utilised two different systems of version control, i.e., it utilized centralised version control mixed with distributed version control. In centralised version control systems (CVCS), there exists a single copy of the project on a server, and any changes made by users are committed to this copy. Each change made by users can be committed and then other users can update their copy of the project in which the changes are reflected. This can allow for real-time editing of documents. A simplified diagram of the centralised version control system is shown in Figure 2.

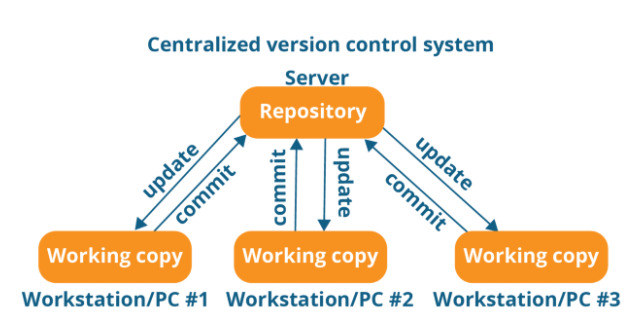


Figure 2: Centralised Version Control System Diagram (Lubański 2019)

To keep track of the different versions of documents, DVCS was used, in which each different version of the document was committed to the repository.

## Version Control Tools

Microsoft Teams was used as the CVCS repository and allowed for easy collaboration on documents. It facilitated real-time editing of the documents from a single centralised server. Each time the document versions were updated, they were pushed to the DVCS repository (GitHub) such that each version can be easily accessed. The use of both version control systems provided added assurance, in the event that one of the systems failed, the documents would remain on the other system.

## Version Control Scheme

**Version Control Method:** Centralised/Distributed

**Notation:** DocumentNameAbbrev\_vA.B

**Parameters:**

1. DocumentName: The abbreviated document name
2. vA.B: Version Number
3. **A:** Incremented each time the document is updated to a complete, finalised version. The initial value is 0. Each time the A value is incremented, the B value is reset to 0.
4. **B:** Incremented each time significant changes are made to the document.

**Example:** VCS\_v1.0

# References

Lubański, Mateusz. 2019. "Centralized vs Distributed Version Control System." *FAUN Publication.* April 30. Accessed September 30, 2021. https://faun.pub/centralized-vs-distributed-version-control-systems-a135091299f0.