# COMP3000 Computing Project

## 2024/2025

### Project Title

Janus Version Control System

### Links

Source code:https://github.com/benjaminsanderswyatt/Janus-Version-Control

### Project Vision

*Outline your initial project vision here. Setting out your project vision at the start does not set it in stone, it can change as you develop your ideas, but you must provide a starting point.*

*Follow the format suggested here* [*https://martinfowler.com/articles/lean-inception/write-product-vision.html*](https://martinfowler.com/articles/lean-inception/write-product-vision.html)

*For [who?]*

Software developers who need an efficient and reliable way to manage code changes, store multiple versions, and access projects across various devices.

*Whose [problem]*

*The [name of your product]*

*Is a [type of product]*

*That [what are the key reasons]*

*No need for the different from or our product.*

For project developers

Whose problem is that they need an easy-to-use system to manage changes, store previous versions and access projects across multiple devices.

The Janus Version Control System

Is a version control system

That combines a command line interface (CLI) for local version control with a web app that allows developers to remotely manage and access their project from many different devices. This includes storing multiple versions of projects, viewing/reverting to past versions and uploading the project to the web app to be downloaded at a later time.

### Risk Plan

|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation |
| Feature Creep | Deviating from the planned features, which delays projects development. | Adhere to my planned sprints, leaving additional features for if time allows development |
| Ease of use | That the product is unintuitive or difficult for users to understand and use. | Utilising user feedback throughout the project to understand what areas might needs improvement. Clearly document how the product should be used, ensuring its readability. |
| Scalability | The product fails to handle increased loads effectively. | Plan early on the systems to be used, while testing larger loads throughout. |
| Performance | Slow processing or a lot of user traffic can lower performance making the product effectively unusable. | Perform stress testing to identify bottlenecks in the product. |
| Security Risks | An unauthorized user gains access to private data. | Implement trusted authentication and encryption for the web app. As well as following best practices for data storage and transfer. |
| Data Loss | Loss of version histories on the data, violating data integrity. | Automatically backup the databases data and implement a routine to recovery from the backup. |
| Licensing Issues | The project violates software licenses. | Ensure all libraries and dependencies used are compliant within the licenses. |
| Legal Issues | The product might deal with sensitive data and must comply with regulations such as GDPR. | Encrypt all stored and transferred data. Allow users to delete their stored data. |
| Legal Liability | Users can store copyrighted or illegal data through our product. | Have a clear term of service that outline the responsibilities the system and the users hold about the data stored. |
| Platform Compatibility | Different platforms might cause the product to fail. | Use cross-platform frameworks to develop the project, while testing the product on multiple platforms. |
| Concurrency Issues | Users attempt to modify or access data simultaneously, causing conflicts. | Implement systems to detect and handle conflicts (e.g. Optimistic database locking). |
| Data Consistency | The user attempts to upload their data to the web app, but an issue occurs (e.g. they lose internet connection or data conflicts occur). | Transactions are used to ensure that if any part of the process fails the transaction is rolled back. |
| Integration Issues | Unexpected problems occur while integrating separate parts of the system (Command line interface, web app & database). | Plan how each part will communicate with each other, while testing regularly for any issues. |
| Technical Issues | Problems such as bugs and crashes occur unexpectedly throughout, delaying the development. | Organise and conduct unit, integration and system testing throughout development, while accounting for delays during planning. |

### Proposed Gantt chart

**Key:**

**Light green:** Research & Planning

**Blue:** Command line interface

**Red:** Website

**Yellow:** Database

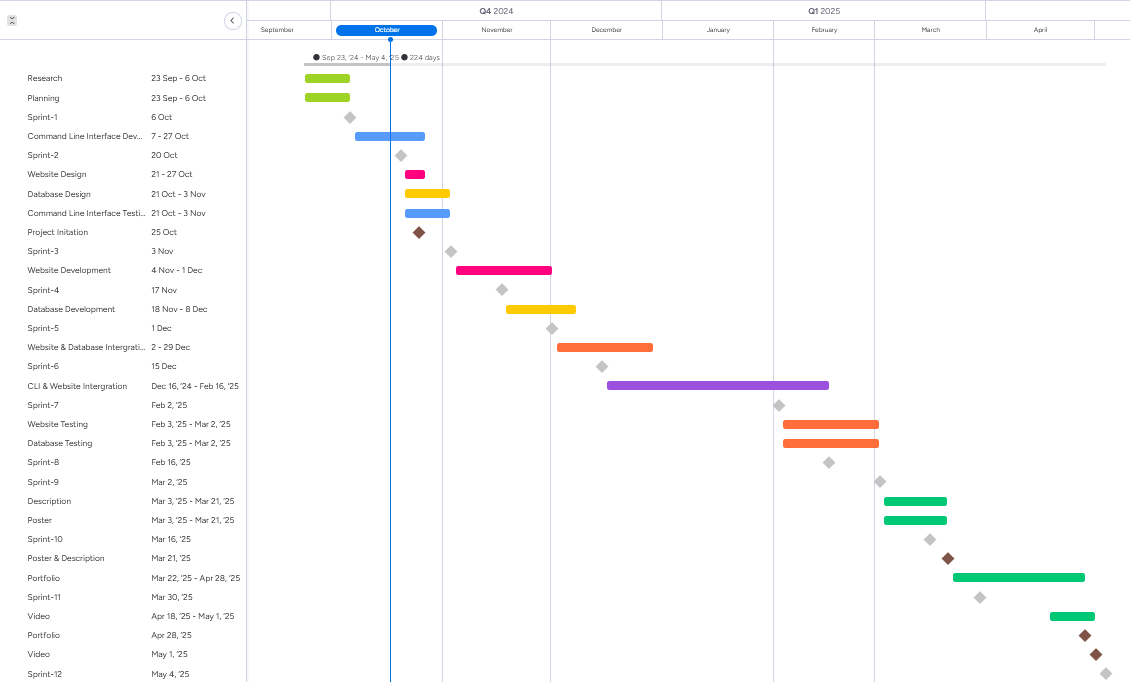
**Orange:** Website & database integration

**Purple:** Command line interface & website integration

**Dark green:** Submissions

**Grey Diamond:** End of sprint

**Brown Diamond:** Submission points



### Keywords

* Version Control
* Command Line Interface
* Web App
* Database
* Project Management
* Commit History
* Backup
* Integrity
* Cloud
* Cross Platform
* Version History