# Construction Phase

## Iteration 1

12/09/15 – 17/09/15:

Once the members met for the first time, a thorough overview of the program design was collaborated on a whiteboard, producing the UML diagrams. After breaking down the design, Thomas then created the database and populated it with dummy data, and performed the necessary checks to ensure the integrity of the database.

While the database was being configured for the project, each member was assigned to write the skeleton of the components previously identified in the overview:

* Ivana: created the stubs for the VersionRecord class.
* Josh: created the stubs for the FileRecord class and began to work on the FileLib class so that it was ready for use in a range of classes.
* Phil E: created the stubs for the FileArchiver class.

The members went home later this day and were assigned tasks to continue on with. Members collaborated over Skype, keeping up to date with one another.

* Thomas completed the code responsible for connecting the FileArchiver class to the database.
* Phil E completed the base functionality of the FileRec and FileArchiver classes. At this stage, the backend of the program was able to handle the creation of the blob data given a file and retrieve it back to some local file system, building off of FileRec, which stores the actual data that FileArchiver passes to/from the database.
* Ivana began to create the base GUI, while Josh was working on FileLib and CPP unit tests for it.
* On completion of FileLib and its tests, Josh and Ivana collaborated to complete the base GUI, which on completion used the functionality produced by Thomas and Phil. The GUI was tested completely functional in terms of displaying what was in the database.
* Phil M and Nicholas collaborated to create the diagrams that were outlined during the initial meeting (where the UML diagrams were drawn up as a group). The integrity of each diagram was verified as a group post-production.

## Iteration 2

18/09/15 – 21/09/15:

During the implementation of the version control (VersionRecord and related components) in regards to the design of the code and the database, we came to the realisation that we can design it more concisely. So, the database design was refactored in the sense that redundant functionality was omitted.

* Thomas created an ERD of the new design. Josh completed FileLib at this point, but some of the functionality was omitted in source files relying on it in the end, as better implementations became apparent.
* Thomas and Phil E worked on refactoring the definitions of VersionRecord.
* Ivana and Josh worked on creating and connecting functionality that wraps the new logic of VersionRecord, which was completed.
* Nicholas worked on compiling notes and screenshots taken by the other members and himself.
* Phil M worked on creating the sequence diagrams of the new design.

21/09/15 – 27/09/15:

At this stage, most of the project has been completed. There are some important aspects of the version control to be implemented, but overall, the project is close to completion.

* Thomas created functionality responsible for file/data compression, which also includes the side-functionality of temporary storage being created during the compression/decompression phase of the interaction between the client and the server/database.
* Ivana and Josh completed the functional requirements of the GUI. All buttons slot into their related functions, and all probable user cases have been identified and handled thoroughly. Most dialogs notifying users of error have been completed.
* Phil E and Josh collaborated to write CPP unit tests for the functional requirements of the backend. Completed and tested thoroughly.  
    
  Josh noticed some logic errors associated with version purging, and notified Thomas to fix. It was fixed in a matter of minutes.

Phil E noticed a troublesome segmentation fault in a case where you close the window without doing anything when blackbox testing the GUI, so Josh and Ivana debugged to find a fault in the destructor of the MyWindow class that allowed deletion of a form that hadn’t been opened. This was fixed promptly.

* Ivana applied final dialogs for user-error cases in the form of message dialogs. The GUI is now complete in all aspects, apart from one core functionality (old version purging). The stub and button have been interconnected as required, encapsulating pseudo-code of the functionality in VersionRecord responsible for the purging.
* Thomas finalized his part of the iteration by cleaning up code related to compression (CompressionUtils) and also fixed it to use gzip instead of zip.
* Josh, Phil E, and Ivana went through and did a final round of testing while also cleaning up useless comments and redundant code.
* The group was notified over Skype of the progress made: almost complete.
* Nicholas and Phil M are compiling what is required for the report, and begin writing it up.

## Iteration 3

The project is practically complete: final touching up and functional testing to finish off, along with the addition of the final component for the project overall, which is also core functionality of the GUI. That is, MyWindow::SetReferenceVersion (use of VersionRecs::PurgeOldVersions in conjunction with Qt components) has to be completed. Finally, the report needs to be worked on.

27/09/15 – 29/09/15:

* Thomas identified an insertion issue with the database. This was fixed promptly. Thomas then finalised the functionality responsible for purging a given number of old version.
* Josh and Ivana wrapped the purging functionality that Thomas wrote up to work within the GUI and its components.
* Phil E requested that Josh adds functionality to FileLib (as he is responsible for the class) for use in the backend CPP unit tests and main source file of the program to setup and destroy temporary directories used for the compression/decompression phase when interacting with the database. Phil E wrote up the prototype to outline what it should do, and Josh implemented it into FileLib as needed.
* The group as a whole was notified over Skype of the completion of the design and code aspect of the project. The group has confirmed the integrity of the overall project, and the report can begin to be written.
* Nicholas and Phil M have working on the report during this time.
* All other members have now been allocated sub-tasks for the report: collaborating to finalise.