CSCI222

Assignment 1

Report

# Overview

*A brief overview of the project based on information in this assignment specification and details of the actual application. This overview should not be created by simply cutting and pasting text supplied. It should be written by the members of the group and provide a retrospective view of the project on its completion. It should be “complete” in the sense that it explains the project to an external assessor who has not seen any of the assignment details.*

# Program Presentation

*A brief presentation that illustrates the actual implemented product. This presentation should combine text commentary with information captured from actual execution of programs (as screen shots or as captured text inputs and outputs). This presentation should clarify which of the functional requirements have been successfully implemented.*

## Functional Requirements

**TODO:** Probably add screenshot of final working instance of these. The text can probably be written beforehand.

Create Initial Archive (completed)

Detect Changes (completed)

Save Modified Version (completed)

Display a summary of versions in storage (completed)

Retrieve chosen version (completed)

Show comment associated with version (completed)

Use compression (completed)

Incremental changes (completed)

Discard old (completed)

# Group

*A tabular summary of the group structure identifying group members, the roles that they filled, the artifacts that they successfully delivered.*

|  |  |
| --- | --- |
| **Role** | **Assignee** |
| Manager (1) | Nicholas Morgan |
| Lead Designer (1) | Josh Coleman |
| Lead Implementer (1) | Phil Edwards |
| Designer (\*) | many |
| Data Persistence Specialist (1) | Thomas Nixon |
| Systems integration and systems test (\*) | Ivana Ozakovic, Phil Edwards |
| Documentation (\*) | many |
| Implementer (\*) | Phil Edwards, Thomas Nixon, Ivana Ozakovic, Josh Coleman |
| Document Backup Maintainer (1) | Thomas Nixon |

## Deliverables of Members

### Thomas Nixon

* Researched the benefits of using MySQL
* Designed new database layout
* Created database diagrams
* Wrote SQL code for database
* Created the SQL statements and C++ code required for the program
* Created initial document in Google Docs for outlining responsibilities and requirements
* Defined layout of classes and member functions and their interaction (With Phil Edwards)
* Implemented VersionRecord member function code
* Moved documents from Google Docs to Git
* Wrote code to handle compression of files and implemented it in required functions

### Phil Edwards

* Created GitHub suppository for project
* Created initial class files
* Created the Netbeans project
* Wrote coding standard document
* Fixed indentation issues across Netbeans project
* Moved database connection to a static function in a class
* Add dependency for MySQL to Netbeans project
* Fixed header guard bug
* Added murmur hash to implementation
* Implemented FileRecord
* Implemented FileArchiver
* Implemented RetrieveVersionRecord in VersionRecord
* Fixed Bugs VersionRecord
* Created Wiki for Git
* Created TODO document
* Wrote Git Primer for other members to refer to
* Defined layout of classes and member functions and their interaction (With Thomas Nixon)
* Created a modified version of murmur hash function to read from file.
* Wrote test code for functions in FileArchiver & FileRecord
* Added logging to program
* Made some modifications to database
* Created MySQL database server setup for other group members to use

### Ivana Ozakovic

* Created GUI main window and dialogs in QtBuilder.
* Created all the GUI functionality in collaboration with Josh.
* Took GUI screenshots for the report.
* Created Program Presentation section for the report.
* Created GUI Implementation and Planning section for the report.
* Declared data members and functions in the VersionRecord.h.
* Created VersionRecord.cpp file and set up stubs for declared functions.

### Josh Coleman

* Created all functionality for FileLib.
* Created all CppUnit tests for FileLib.
* Took images of FileLib code (both header/stubs and source/definitions).
* Took images of FileLib CppUnit tests.
* Created the table model for the table view in the GUI.
* Worked on all GUI table/button functionality (in collaboration with Ivana).

### Nicholas Morgan

* Meeting Reports
* Documentation
  + Elements and unit testing procedure
  + Supporting code samples and listings
  + Version Management – initial
  + Detailed meeting report - revision

### Phil M

# Design & Implementation

*A summary of your group’s work on design and the implementation plan. This should cover: any reworking of the proposed implementation classes and give details of decisions relating to data persistence and user interface issues. UML modeling diagrams should be used to illustrate*

*design decisions. If your group decides on a different implementation plan, with different iterations, you should give details and justification.*

* *Details of the construction phase. This part of the report should clarify the work done in each iteration.*
* *Summarize the new elements added and the extensions to existing implementation elements. Include brief details of unit testing procedures used to verify new elements prior to their commitment to the project (this does not mean list unit test code and test outputs, simply identify the additional tests created by individual implementers).*
* *Summarize data from defect and integration reports created by the systems integrator.*
* *Provide evidence for the appropriate use of version control software; this would typically take the form of excerpts from subversion’s logs of commit operations.*
* *Subversion statistical reports, showing overall contributions by different members, could be included in the report on the final iteration. N.B. some practice use of the subversion code management system is a requirement of this exercise. Of course it’s overkill here. The intention is that you practice the use of such technology on something fairly trivial before you need to use it for real as in CSCI321.The important thing is that you gain some practice in the use of a version management system.*

## Design Choices

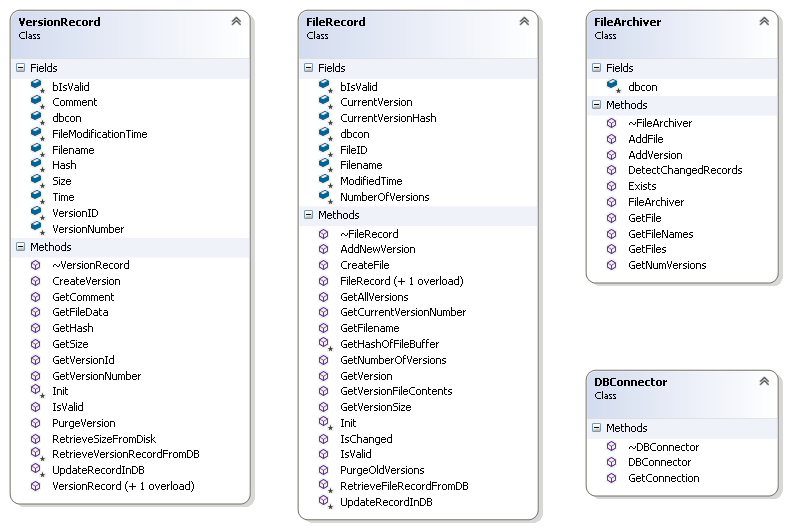
### Database

Diagrams go here

Explanation of why we changed it

### FileArchiver

Diagrams go here



Explanation of why we changed it

## Implementations

### Iteration 1

New elements added

Tests used

Functional Requirements met

### Iteration 2

New elements added

Tests used

Functional Requirements met

### Iteration 3

New elements added

Tests used

Functional Requirements met

## Version Control

How we used it

Setup

Details

Access

Usage Information

# Group Records

*Group meeting records and individual diaries:*

* *There should be a summary detailing the work done at each formal group meeting.*
* *There should be an example agenda, and report from at least one of these meetings.*
* *There should be samples taken from the work diaries of at least two members of the group.*
* *There should be samples from bug logs and testing logs*

## Summary

## Agenda

## Work Diaries Samples

## Bug/Testing Log Samples

# Code

*Code listings for all elements in the final product should be included in an appendix to the main report. Samples should be provided of support code such as the unit test classes created to verify particular application classes.*