Critical Evaluation

The team performed very well during this project even during setbacks faced throughout the project. All members clearly knew their areas of expertise and weakness allowing the team to best utilise individual skills. An area for improvement is during quieter weeks additional training could have been performed ensuring skill sets are more equally spread across team members. During this project members were present throughout however if somebody with a unique essential skillset went ill it would have caused heavy delays.

The project could have been improved in two major ways. Firstly the final product did not have any unit tests built for it. Initially we were planning on using test driven development; however our lack of experience with the technology meant whilst tests would pass the program would not function correctly.

Following this the team agreed additional prototyping would have paid dividends. Our initial prototypes proved concepts such as using window builder to create GUIs but complete additional steps such as refreshing the displayed data. This lead to large deviations from the original plan which assumed certain functionalities that could have been avoided with more rigorous prototyping.

From this extra prototyping new unit tests should be built, a common issue during implementation was regressions which went largely undetected from the lack of unit tests. Whilst unit tests would have taken additional time to produce, the time saved would have paid dividends.

The team developed an understanding of the need for a well-developed project plan. By completing research and identifying risks in new technology additional testing and training can take place. This prevents large design changes, unnecessary refactoring and bug introduction. In larger projects with multiple teams these changes would significantly impact on other teams who may depend on the old incorrect design.

Finally whilst within more complex methods it is worth breaking them into sequence diagrams. Whilst the implementation may seem trivial and obvious to all people everyone will still approach that algorithm differently. This leads to people assuming the algorithms behaves in a certain manner as it is not documented and leads to subsequent bugs stemming from those assumptions.