1 Group Report

1.1 Summary

At the beginning of the project, we as a team were tasked with producing a website for 'Last.fm' called 'Scrobble Exchange'. This would be a market-based game involving the artist data that last.fm store and provide freely for people to use, and we decided that the best way to achieve this would be to create a game where you could buy and sell artists, similar to a stock market for musicians. Below is a summary of the failures and successes of this project, the lessons that we have learned, and the individual work that each team member completed.

1.2 Successes

- The design of the game went successfully. Although we spent a long time on discussing game mechanics and deciding on how to make the game fun, the final idea and vision that we came up with was very compelling and interesting.

- All the components of the game were successfully implemented - the frontend, the API, DATM and the analytics module. Although some of these didn't have all of the initially planned features, at the end of the project they were all fully realized and working.

- In the addition to the above, the core mechanics of the game were also implemented, as planned initially (buying and selling).

- The scalability of the system was implemented as planned - all of the major components are separated and designed in such a way as to allow for multiple people to be using the system at the same time.

1.3 Failures

- In many situations, we failed to communicate. By this I mean that after the initial design stage, we didn't meet as often as we should have, and didn't communicate what each of us was working on very well. This meant that code was not written and people were often confused as to what other people were doing, and what they should themselves be doing.

- We failed to manage our time properly. The code was mostly written and submitted to the repository in short bursts rather than at a steady pace, which would probably have been a better option. Indeed, we didn't stick to the schedule we had planned initially, which lead to long-term issues in being able to integrate the various code bases together, and getting the entire program working.

- We failed to test our work extensively. Although we had initially planned to unit test all the components of the game, we didn't get past 'printf debugging' for much of it, which is clearly not a good level of testing.

- The game mechanics had a lot of content cut in the interests of simplification and saving time. Arguably, the game is not as good as following the entirety of the original design would have made it. Had we managed our time better, it is likely that we would have been able to implement all of these 'wants' rather than 'needs'.

- The UI turned out far more minimalistic than planned - we expected to have more interaction with last.fm on the website, such as linking to their radio services.

1.4 Lessons Learned

- Group Leader - at the beginning of the project, we decided against having a group leader because it would give a single person too much influence within the team. With hindsight, this has turned out to be a bad decision since there was no one to keep the group the schedule and motivate them to keep working and keep coding. Having no leader is only a good decision if everyone on the team is already passionate and motivated, and can either keep themselves working, or are good enough friends/acquaintances that there will be social pressure for them to work. Hence, in any future projects I will ensure that that at least one person is designated to lead the team and take responsibility for this.
- Time management - having a set schedule is no guarantee that this schedule will be kept to. It is necessary to have more (such as a team leader) in order to ensure that any plans are kept to.
- Integration - it is difficult to integrate multiple modules written by multiple people. It works best if people communicate and are able to ask each other questions, and therefore in future joint coding sessions may be a good idea in order to ensure that code gets written swiftly and correctly. Equally, encouraging good documentation of the code and using easy to understand variable names also helps to this.

1.5 Work Undertaken

1.5.1 Amar Sood

Amar Sood was responsible for writing and creating DATM, one of the four major components of the system. This involved a large amount of coding and design, and writing this single-handedly meant that other members of the team were free to work on the other parts of the project. He was also responsible for the documentation and testing of this subsystem.

1.5.2 Joe Bateson

Joe Bateson was responsible for making the front of the frontend of the game, designing and coding the webpages that are displayed to the players. This was half of one of the major components of the system, and therefore a significant amount of work. He also tested and documented his work.

1.5.3 Neil Satra

Neil Satra was responsible for writing the back of the frontend of the game, which would provide an interface between Joe's work and the API. Since without this the players of the game would never be able to receive any information from our database, his work was critical to the success of the project.

1.5.4 Victor Mikhno

Victor Mikhno was responsible for writing the API, which provided the go-between of the frontend and the backend of the game. This was the third major component of the system and without it, it would have been impossible for the frontend to ever serve the user any data. He also wrote the documentation and performed testing on the API.

1.5.5 Guoquiang Liang

Guoquiang Liang was responsible was the analytics module, which provided the calculations for the information we pulled from last.fm. As the fourth major component, this module was important for implementing the mechanics of the game and ensuring that the game was balanced and (conceptually, not practically) difficult to cheat at.

1.5.6 Karolis Dziedzelis

Karolis Dziedzelis was also responsible for the analytics module, and worked together with Guoquiang to define the formulae and mathematical foundation of the game.