Final Report & Program

Team 7A.1

ComPChecker

Name	Contributions	Signature
Thomas Chate	See GitHub for program contributions. Though, it is agreed between the team that work is equal. Final report was written equally.	TChate
Lillie Hogg		LHogg
Luke Jackson		L. Jackson

Account of The Project

This project was started under the administration of six members. In our very first meeting we focused on getting to know each other rather than technical details, learning about hobbies and interests of the team would prove beneficial to creating a system which we are passionate on. Skill sets were also discussed, both technical and softskill, this enabled the team to play to our strengths. After some consideration each member was given a technical role and a non-technical role, such as time 'HR' this role ensured milestones and deadlines were met. Technical roles were given to the member with the most expertise. It was also decided that weekly meetings shall occur, to ensure that members could support each other with the development. A groupchat was established to support this.

The scope and idea of the project was then debated upon, at first we had no suitable idea. However, after consideration we decided upon creating a PC building system to mitigate any possible compatibility errors between PC hardware. Initially, it was proposed that the system was to be created as a website, however, after skill analysis only a few members had experience with developing successful websites. Therefore, Java was chosen, a language familiar to all. A thorough risk assessment was also created at this stage, this would prove to be beneficial later.At this stage, one member wanted to begin coding. However, we all agreed to wait until other sections had been completed, such as design documentation. This would prevent any adaptive changes being made to the system.

The team then begun working on the SRS and dubbed the development name of the system as 'ComPChecker'. At this stage prospective users, or users would have used similar systems previously, were interviewed to start outlining user requirements. This stage highlighted the main demands from users and as a team we were able to prioritise them to fit within the timeframe and the scope of the project. Interviews were also conducted on several users so more open questions could be asked and requirements could be clarified further. Retrospectively, a greater number of users should have been interviewed, requirements could have been further prioritised by the user. It was at this stage a team issue was raised, some members of the team were adding in features they thought was beneficial. However, this led to the team being on the wrong 'page', with conflicting views over what should be developed or what should be prioritised. To resolve this issue, an emergency team-meeting was held ensuring that all team members were focused on the task and understood the scope of the project. It was also during this stage were the amount of effort each put in was noticeable, some members only wanted to pass while others wanted to achieve the highest possible grade. This was noticed in the quality of work produced, but due to the established system of checking others work errors could be resolved and with a team deadline of 24 hours prior to the actual deadline it allowed for some confidence in the work.

The next stage, design, was one of the weakest submissions as a team. On the first meeting of this stage, only 50% of members attended. This led to a situation where members were given a task without their input, this was done to speed up the process. However, one member complained about their task so some allocations had to be changed. However, the lack of attendance of the meeting was also combined with the lack of attendance of lectures and seminars. This led to situations where a member had to produce a diagram for a lecture they never attended, this meant certain team members had to support them to ensure a

good document was produced. Work wise, initially we had decided on a pipe and filter architecture as we thought it represented the system. However, after considerations and support a three layered architecture was found to match the domain of the system. As stated earlier this stage was particularly weak. This was partly due to the deadline being set over the Christmas holidays, this required members to work without being reminded by other members. However, several failed to do so and the internal soft deadline being missed by many. Some of the sections were not uploaded until few hours before the deadline, this gave the team little team to ensure accuracy and consistency across the sections.

The next deadline, prototype, proved to be a pivotal turning point in the team. As mentioned previously, work from a few individuals was lacking. This lead to three members taking on more work to ensure the deadline would be met. One member failed to accept the GitHub repository request and became absent from seminars, meetings and lectures. After some difficult discussions; Thomas, Luke, and Lillie decided it would be fair if they should continue as a separate team. As expected, the other side thought this was unfair but with mentoring from Claudia an agreement was reached. The team shall split into two, working on the same project brief. This lead to issues, when we split we lost many of our strongest coders. Tom and Lillie had to quickly learn advanced Java skills to support Luke in the development of the system. After this technical barrier was overcome, the project seemed to be working easier than as a team of 6. Our team of 3 were more dedicated to ensuring high-quality work was produced and this turned into friendship between the members. We also had issues with GitHub when Luke branched the master repository but made major changes so when a pull-request was made there were several conflicts. Luke had to manually sort these out, from this point we all agreed to work from the master and frequently update the project.

We then begun working towards the testing milestone alongside developing the system. It was quickly noted that due to the workforce being 50% compared to normal; the scope of the project needed to be adjusted. Some requirements, such as changing display language, were seen as not necessary at this stage and could be added in later during adaptive maintenance. During this stage our database was hacked. Our database was running on a Raspberry Pi along with other projects, due to the security not being strong enough on the server our data was compromised. But due to future planning, as set out in the risk assessment, a 3 day old backup was available and used to restore the system. Additionally, more security features were added to prevent this happening again. This milestone faced several delays, this was mainly due to the workload for other commitments, one member had 3 deadlines for the same week. Now a team of three work is harder to share and all members must work to their full efforts.

For the final submission over the Easter break the team worked on finishing the piece of software. Due to the scope of the project being reduced some aspects were not added (as stated in the risk assessment), to keep track of this we used the GitHub project tool. This tool also allowed us to assign to tasks to members and track the progress. During this stage the team would have benefitted from reflecting on the stated requirements; rather than "guessing".

;

Issues Throughout Project

Throughout the duration of this project the team encountered several issues within the production of the ComPChecker system. Thomas and Lillie faced issues when running the database, they found that it was not possible for the database to constantly run on the VM provided by the university, and this was confirmed by the database lecturers. Thomas instead resolved this issue by running the database from his housemates Raspberry Pi, which ran with no issues expect from the security breach.

A major issue that the group as a whole, encountered was with GitHub. As certain members of the team had forked their own repository and were then several commits behind the master due to lack of work and communication with the team, it meant that when they committed changes to the master it would remove the current and updated code, and replace it with code which was no longer relevant to the system. This meant that members of the team would then need to revert their commits, which proved to be quite difficult and sometimes resulted in losing major parts of the code.

When conducting meetings, the group encountered disagreements with how the system should be written. Members of the group believed it should be written as a layered interface, while other members of team believed that the system at the current stage should be coded within a number of classes to prevent confusion and enable those less able at Java to be able to contribute more to the application development.

Conclusion

Overall, the project was eventful, raising both technical and non-technical problems. But the greatest learning points cames from the most 'negative' events. The final team learnt how to deal with team divide and how to recover from it, even if the result wasn't one we hoped for. As a team we have become more adaptable, supporting each other and learning new skills throughout the process.

The GitHub repository is located at https://github.com/lilliekhogg/ComPChecker
This repository was forked from another repository before the team split.